Complexity

Intelligent Methods for Large Scale System Operation and Management

Lead Guest Editor: Rui Wang
Guest Editors: Shi Cheng, Guohua Wu, and Shangce Gao

Intelligent Methods for Large Scale System Operation and Management

Complexity

Intelligent Methods for Large Scale System Operation and Management

Lead Guest Editor: Rui Wang
Guest Editors: Shi Cheng, Guohua Wu, and Shangce Gao

Contents

Contents

Hindawi
Complexity
Volume 2021, Article ID 9850964, 17 pages
https://doi.org/10.1155/2021/9850964

Corrigendum

Corrigendum to "A Novel MILP Model for the Production, Lot Sizing, and Scheduling of Automotive Plastic Components on Parallel Flexible Injection Machines with Setup Common Operators"

Beatriz Andres, Eduardo Guzman, and Raul Poler

Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València (UPV), Calle Alarcón 03801 Alcoy, Alicante, Spain

Correspondence should be addressed to Beatriz Andres; bandres@cigip.upv.es

Received 5 August 2021; Accepted 5 August 2021; Published 18 November 2021

In the article titled "A Novel MILP Model for the Production, Lot Sizing, and Scheduling of Automotive Plastic Components on Parallel Flexible Injection Machines with Setup Common Operators" [1], the authors identified errors in the datasets used in the study. Specifically, there was an error in the *ci* (inventory cost) parameter which uses decimal values; however, the python code used it as an integer. The datasets have therefore been corrected in Tables 6–11, and the corresponding sections in the results section have been updated to reflect this. A revised data availability statement has also been provided. The authors confirm that the results and conclusions are the same as reported in the original article despite the revised datasets.

The corrected article is as follows:

In this article, a mixed integer linear program (MILP) model is proposed for the production, lot sizing, and scheduling of automotive plastic components to minimize the setup, inventory, stockout, and backorder costs, by taking into account injection molds as the main index to schedule on parallel flexible injection machines. The proposed MILP considers the minimum and maximum inventory capacities and penalizes stockout. A relevant characteristic of the modeled problem is the dependence between mold setups to produce plastic components. The lot-sizing and scheduling problem solution results in the assignment of molds to machines during a specific time period and in the calculation of the number of components to be produced, which is often called lot size, following a sequence-dependent setup time. Depending on the machine on which the mold is setup, the number of units to be produced will be distinct because machines differ from one another. The stock coverage, defined in demand days, is also included in the MILP to avoid backorders, which is highly penalized in the automotive supply chain. Added to this, the proposed model is extended by considering setup common operators to respond to and fulfill the constraints that appear in automotive plastic enterprises. In this regard, the MILP presented solves a lot-sizing and scheduling problem, emerged in a second-tier supplier of a real automotive supply chain. Finally, this article validates the MILP by performing experiments with different sized instances, including small, medium, and large. The large-sized dataset is characterized by replicating the amount of data used in the real enterprise, which is the object of this study. The goodness of the model is evaluated with the computational time and the deviation of the obtained results as regards the optimal solution.

1. Introduction

Production planning, sequencing, and scheduling are key operations performed by enterprises, and any circumstances or events that affect them strongly influence the supply chain operation in which they are embedded. All these three

planning levels are characterized by the decision-making time horizon in accordance with three decision-making levels: strategical, tactical, and operational. Thus, production planning is set at the strategic level by considering families of products, while sequencing and scheduling are set at a more operational decision-making level. Accordingly, Gujjula et al. [1] proposes the following differentiation between these two concepts: (i) production scheduling deals with the assignment of production orders to production intervals with a short planning horizon and specific time periods and (ii) production sequencing deals with the sequence of production orders for each production interval.

Our aim is to solve a lot-sizing and scheduling problem with a sequence-dependent setup on parallel flexible machines. To this end, a mixed integer linear program (MILP) model is proposed to minimize the setup, inventory, stockout, and backorder costs by taking into account injection molds as the main index to schedule on parallel flexible injection machines. We also consider setup common operators to extend the proposed basic MILP model.

This article focuses on the specific production lot-sizing and scheduling problem (LSSP) in parallel flexible machines. According to Kim et al. [2], the LSSP deals with the minimization of production and inventory costs by simultaneously optimizing lot sizes and production schedule. The authors also refer to small and big bucket models depending on the number of allowed setups. Accordingly, production scheduling assigns production orders to production intervals with a short-term planning horizon lasting several days or shifts. Moreover, the assignment in the scheduling process has an implicit sequence for each production shift. Therefore, the main goal is to identify the time period when to produce, the quantity to produce as the lot size in units or timeslots, and the production sequence required to meet demand and to avoid backorders and stockouts.

As enterprises are seen as complex systems in operations management, their operation mechanisms are difficult to manage. This difficulty further increases because enterprises belong to a supply network system, in which the complexity and relationships with external actors are latent. Current global market conditions and constant changes in the supply chain environment render enterprises as complex systems. Moreover, researchers have to bear in mind that market consumers are currently used to acquire highly personalized products, which is known as mass customization with short development periods ("time to market").

In the last few years, novel technologies have been increasingly used, such as cloud computing, big data, artificial intelligence, and machine learning. This has become a trend that has boosted companies to transform their way of operating at enterprise and supply chain levels. The result of this digital transformation has been coined as Industry 4.0 [3]. Nevertheless, the application of Industry 4.0 to small- and medium-sized enterprises (SMEs) is not as idealistic as the Industry 4.0 definition indicates. The literature includes different studies on implementing Industry 4.0 technologies into different sectors. The present work focuses on the literature review by Echchakoui and Barka [4], which studies the impact that Industry 4.0 has on the plastics industry. The relevance of the work by Echchakoui and Barka [4] is aligned with the present article because we develop MILP to support the LSSP in a company that belongs to the plastic sector by particularly focusing on the injection of plastic automotive components. In the aforementioned literature review, the authors highlight that implementing Industry 4.0 research into the plastics industry is still in its initial stages, but research in this field is growing.

The origin of this article lies in the H2020 Project Cloud Collaborative Manufacturing Networks (C2NET) [5], whose research focuses on providing a cloud platform, in which a set of tools and technologies are embedded to support the following: (i) data management and interoperability; (ii) the optimization of plans at enterprise and supply chain levels; and (iii) the integration of collaborative processes among supply network members. The C2NET cloud platform is built according to an open-source philosophy, which makes it affordable and easy to use by SMEs in terms of both monetary and expertise or knowledge required for its use. The optimization module contains advanced optimization models and algorithms to support and calculate replenishment, production, and delivery plans at both the enterprise and network levels. The calculation of plans at the enterprise level is characterized by only considering the resources and data from a single enterprise, while the calculation of plans at the network level uses the constraints, resources, and data from two supply chain enterprises, or more, as input data. When solving replenishment, production, and delivery plans from a collaborative network perspective, the output data of one enterprise plan are used as the input data of another enterprise plan. If we consider two enterprises from a network, A and B, where A is the supplier and B is the manufacturer, the replenishment plan of company B is constrained by the production plan of company A. Hence, information is exchanged and a loop plan is calculated until the materials required by company B coincide with the materials that can be produced by company A. On the C2NET cloud platform (CPL), the negotiation loop is operated by the collaborative module (COT), the plan calculation is managed by the optimization module (OPT), and data exchange and information interoperability are handled by the data collection framework module (DCF) [6]. The operation and management of the complex large-scale systems that characterize enterprises are covered by the models and tools developed in the CLP with the help of embedded intelligent methods.

Different enterprises from diverse sectors were involved in the C2NET project to validate and test the generated results. The enterprise study object of this article is the automotive industry, which is included in one of the industrial pilots. The automotive supply chain is characterized by the need to perform flexible manufacturing to meet the demand of the original equipment manufacturer (OEM) in terms of delivery time, scheduling, and lot size in a just-in-time (JIT) production system [7]. When addressing the LSSP in the automotive industry, suppliers are highly penalized when the components supplied to the OEM are delayed.

Thus, the automotive components' manufacturers have to manage minimum and coverage stocks to avoid potential penalizations. The coverage goal is set in demand days so that suppliers have to produce at least 3 days of demand in advance (commonly used in the automotive industry) to stock production, as advances to avoid delaying demand because it could imply stopping the automotive assembly line in OEM facilities.

The main objective of this article is to model a real problem from a second-tier supplier in the automotive supply chain and to solve it in a reasonable computation time. Accordingly, a lot-sizing and sequencing model is hereafter provided to respond to the requirements of a second-tier supplier from an automotive supply chain. The proposed optimization model is embedded in the OPT module of the C2NET CPL. A novel MILP model for automotive plastic components' production lot-sizing and scheduling on parallel flexible injection machines is proposed with setup common operators. The MILP model was implemented in Pyomo [9], used as an extensible python-based open-source optimization modeling language for linear programming and nonlinear programming, among others. Finally, a complex large-scale problem is addressed to deal with the scheduling plan of an automotive components' manufacturer.

The reminder of this document is organized as follows. Section 2 presents a literature review to gain insight into works that have addressed similar type problems and to justify this study's contribution. Section 3 describes and analyzes the problem to be solved. Section 4 presents the novel MILP model for the automotive plastic components' production lot-sizing and scheduling on parallel flexible injection machines. We take this MILP as the base model. Section 5 puts forward an extension of the base model, which contemplates common setup operators in the proposed LSSP to provide a more realistic perspective to the base model by taking into account the studied plastic components' injection industry. Section 6 offers comprehensive numerical experimentation by considering different data sizes for both proposed models, including small, medium, and large datasets. This article is concluded by discussing the findings obtained with the addressed problem.

2. Related Studies

A large amount of research has been conducted into different LSSP characteristics, but very few studies present optimization approaches that combine or integrate LSSP characteristics, that is, lot sizing and scheduling. According to Ríos-Solís et al. [10], the product-part-mold-machine (PPMM) problem and the part-mold-machine (PMM) problem study this combination of approaches and indicate that this problem type is scarcely addressed in the literature. Ríos-Solís et al. [10] defined products as final products and parts as components that derive from the bill-of-materials parts explosion. Furthermore, Ríos-Solís et al. [10] classified the PPMM as bilevel capitated LSSP. Accordingly, the first part of the PPMM and PMM lot-sizing and scheduling problems seeks to determine the optimal size of either a lot of products in the PPMM approach or the parts in the PMM approach. Both approaches seek to assign parts or pieces to molds and molds to the machines. The second part is involved in the scheduling that aims to determine the order of processing molds in machines during each planning time period.

Studies such as Ibarra-Rojas et al. [11] addressed the problem of manufacturing parts that are produced in molds and are mounted on machines. This study proposes an MILP that seeks to maximize the production of parts and also determines the batch size of every part and the assignments of parts to mold and machines. An update of this study is that proposed by Ríos-Solís et al. [10], which aims to determine the lot size of a finished product; that is, it determines the number of finished products to be manufactured, the number of parts to be manufactured, and the assignment of the parts to the mold and the mold to the machine. The model of Ríos-Solís et al. [10] also contemplates the feasibility of scheduling molds in machines during each period. However, in their experimental results, they only handle one period of time periods and propose working with multiple time periods as a future research line. Different approaches have been proposed in the literature to optimally solve scheduling in engineering; in this regard, Li et al. [12] highlighted the importance on the use and application of MILP to deal with the aforementioned problem in the engineering research area.

Studies about LSSP are paid more attention by researchers and companies given their applicability to the real world [13]. In our literature review, we find studies that integrate batch sizing decisions into restrictions for scheduling problems (see Table 1), studies such as that presented by Stadtler [14], in which a combined approach to support a single machine LSSP is proposed. For this purpose, an MILP is formulated to minimize maintenance and inventory setup costs over a planning horizon. Wolosewicz et al. [15] combined production planning and scheduling by proposing an MILP that seeks to determine the lot size for a fixed sequence of operations in the machines, taking into account the times and operating costs. Kim et al. [2] presented a combined approach in an MILP that seeks to minimize the sum of the cost of production, installation, and inventories. James and Almada-Lobo [16] presented an MILP for the scheduling problem and capacitated lot-sizing of a single machine and a parallel machine with sequence-dependent setup times and costs. Meyr and Mann [17] put forward an MILP to simultaneously determine production lot sizes and schedules on nonidentical parallel production lines.

Our model deals with the problem contemplated by Ríos-Solís et al. [10] and Ibarra-Rojas et al. [11]. By continuing with the future research lines indicated by these authors, our study considers many periods when modeling and running experiments to solve the LSSP. The base model herein proposed also bears in mind stock coverage constraints, which are typical in the studied automotive supply chain industry context. The proposed novel MILP also contemplates an objective function

Table 1: Literature review of recent lot-sizing and scheduling problems.

Authors	Objectives	Industrial application	Resolution methods	Solver
[2]	Minimize the sum of production, setup, and inventory costs	Zinc refinery	Heuristic algorithm combining a decomposition scheme with a local search procedure	CPLEX
[11]	Maximize the weighted cost of produced pieces	Automotive, consumer goods, and toys	Decomposition approach	CPLEX 11.2
[14]	Minimize the sum of inventory holding and setup costs in the planning interval	Pharmaceutical		Xpress-MP
[10]	Maximize the profit of finished products	Automotive, consumer goods, and toys	Iterative heuristic based on mathematical programming	Gurobi 6.05
[17]	Minimize inventory holding, sequence-dependent setup, and line-specific production costs of potentially heterogeneous production lines	-	Decomposition-based approach	GLPK 4.44
[15]	Minimize the sum of production, inventory and setup costs	-	Lagrangian heuristic	Xpress-MP

based on the assembly line and allows idle times among molds, which is a fundamental characteristic for real cases and has been ignored by former studies. Finally, the base model is extended by offering a scenario that comes closer to reality by considering another index that represents the workers who change molds and, thus, bears in mind the casuistry associated with the LSSP that refers to the setups of the usual operators.

3. Problem Description

The main aim of this article is to propose a novel MILP model for the production scheduling carried out by an automotive plastic components' manufacturer that acts as a second-tier supplier in the automotive supply chain. Plastic components are produced in molds that are mounted on parallel flexible injection machines. Injection machines shape plastic pellets into automotive semifinished products to then be assembled on an OEM production line. Nevertheless, for industry, studying car components is treated as the second-tier supplier's end products.

According to Ríos-Solís et al. [10], lot-sizing and scheduling plans of mold-injection enterprises entail decision-making to determine (i) the lot size as the amount of components to be produced during a period of time or a number of periods to produce the same component and (ii) the assignment of molds to machines by considering that components can be produced only in specific molds with their shape cavity.

The second-tier supplier herein studied is characterized by having specific molds that produce each automotive component. When two molds are available to produce the same component, these molds involve different processing times given their technical particularities. Each mold can also be setup on different machines to produce the same automotive component, but the same mold mounted on different machines has different production rates depending on the machine on which it is setup. Accordingly, productivity differs depending on the mold and machine assignment (see Figure 1). Moreover, in the automotive sector, two different parts are produced in the same mold, including right- and left-hand parts, which is known as biproduct injection molding [18].

The studied enterprise has 21 injection machines and a set of molds to produce the range of components delivered to the first-tier supplier and finally to the various OEMs that belong to different automotive supply chains, depending on the car brand. The enterprise works three shifts per day over a 5-day week and reserves overtime shifts on the sixth weekday in the event of production not finishing during the normal operation time. Workers are told that they will work overtime periods at least 1 week beforehand.

Molds are changed using cranes and, therefore, a setup time is incurred. A limited number of workers are in charge of changing and mounting molds to the assigned machines. Moreover, these workers, known as setup common operators, work only two of the three shifts that the enterprise arranges.

One of the requirements of the enterprise's study is that, once the mold is setup on a machine, the mold should remain for at least 24 h so as not to saturate the work of the limited setup common operators and to not incur on too many setups because the setup time is estimated between 1 and 3 h, which obviously has an associated setup cost. If a longer production time is needed, the mold is set up during the required time periods without incurring any setup costs. Thus, the modeled problem should contemplate no setup carry-over cases.

Backorders are highly penalized in the automotive supply chain. Therefore, enterprises in the automotive industry use stock coverage, which indicates the number of demand days that stocks can cover. Normally in the automotive industry, stock coverage is set at 3 demand days; for example, for the demand of four units for the next three periods (d_1 = 4 units, d_2 = 4 units, and d_3 = 4 units), stock coverage is defined as 12 units at the end of the first period. Moreover, as warehouses have space limitations, a maximum inventory is considered.

With regard to the demand, the automotive industry updates the demand during each period for the next five

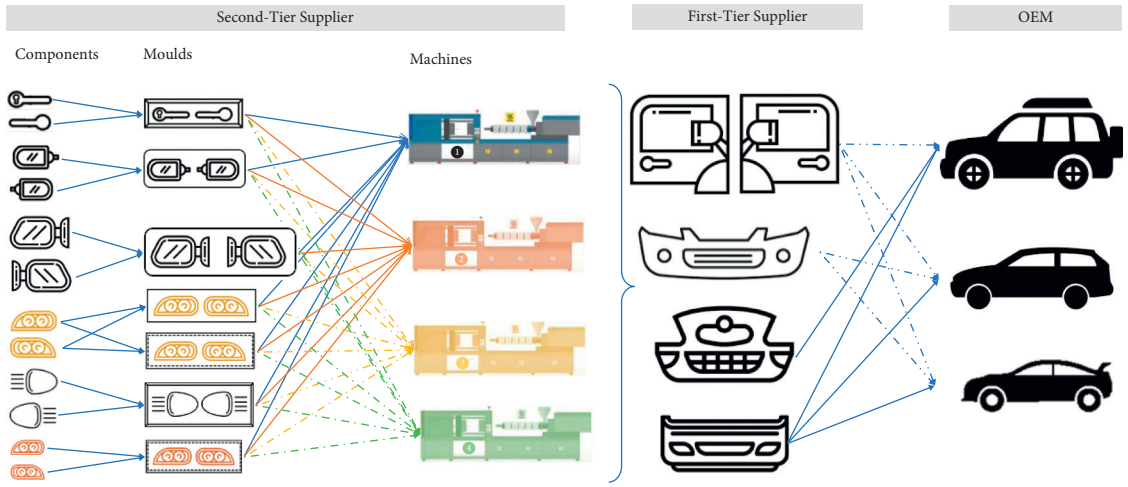


Figure 1: Production scheme on parallel flexible injection machines of plastic components in the automotive supply chain.

frozen periods. The OEM works with considerable demand information (1 year of demand horizon), but only communicates the demand for the next 6 months to the first-tier supplier. Finally, first- and second-tier suppliers normally work to a 3-month demand horizon and with daily periods. The LSSP considers a 21-day horizon.

A detailed flowchart of the processes is presented in Figure 2, from the generation of the customer demand to the final resolution of the second-tier supplier model. Accordingly, in the automotive supply chain, the OEM generates customer orders, according to the final customer demand. The OEM transforms the customer orders into the master production scheduling (MPS), which jointly with bill of materials (BOM) and the inventory availability computes the materials requirement plan (MRP). The OEM demand plan is generated from the MRP and transferred to the first-tier supplier. Then, the first-tier supplier obtains its demand by using a simple bill-of-materials parts explosion given the OEM demand plan. The second-tier supplier estimates the component requirements from the final requirements in a frozen sequence transferred by the first-tier supplier, and the demand plan is generated [8]. Then, the second-tier supplier proceeds to compute the LSSP model with the aim of (i) assigning molds to machines, (ii) scheduling the processing molds in machines during each planning time period, and (iii) calculating the optimal lot size of products. The LSSP is modeled by considering molds to be the main index. The proposed production lot-sizing and scheduling base model has as main inputs the data parameters described in Table 2. Nevertheless, if setup common operators are considered when solving the LSSP, data parameters related with the type of operators have to be taken into account for building the extended version of the base model. Finally, the proposed MILP is solved by the second-tier supplier using the Gurobi solver obtaining optimal or near-optimal solutions of the production LSSP. In a nutshell, the objective of the proposed MILP is to minimize the total costs, namely, setup and inventory costs, penalization costs for coverage stockouts, backorder costs, and tool setup costs, such as the route cost of selecting one machine or another to setup a tool.

4. MILP Model for Lot-Sizing and Scheduling on Parallel Flexible Injection Machines: Notation and Model Formulation

The lot-sizing and scheduling on parallel flexible injection machines problem under study is notated in Table 2, where the main indexes related to machines, tools, products, and periods are notated. The input data parameters are presented, and decision variables are established as the output data of the MILP model for lot-sizing and scheduling on parallel flexible injection machines.

The formulation of the MILP model for lot-sizing and scheduling on parallel flexible injection machines is described next. The objective function minimizes total costs, including setup and inventory costs, penalization costs for coverage stockouts, backorder costs, and tool setup costs, such as the route cost of selecting one machine or another to set up a tool.

$$\text{Min } z = \sum_i \sum_j \sum_t cs_j \cdot SA_{ijt} + \sum_k \sum_t ci_k \cdot \text{INV}_{kt} + \sum_k \sum_t cst_k \cdot ST_{kt} + \sum_k \sum_t cb_k \cdot B_{it} + \sum_i \sum_j \sum_t r_{ij} \cdot cr_{ij} \cdot SA_{ijt}, \quad (1)$$

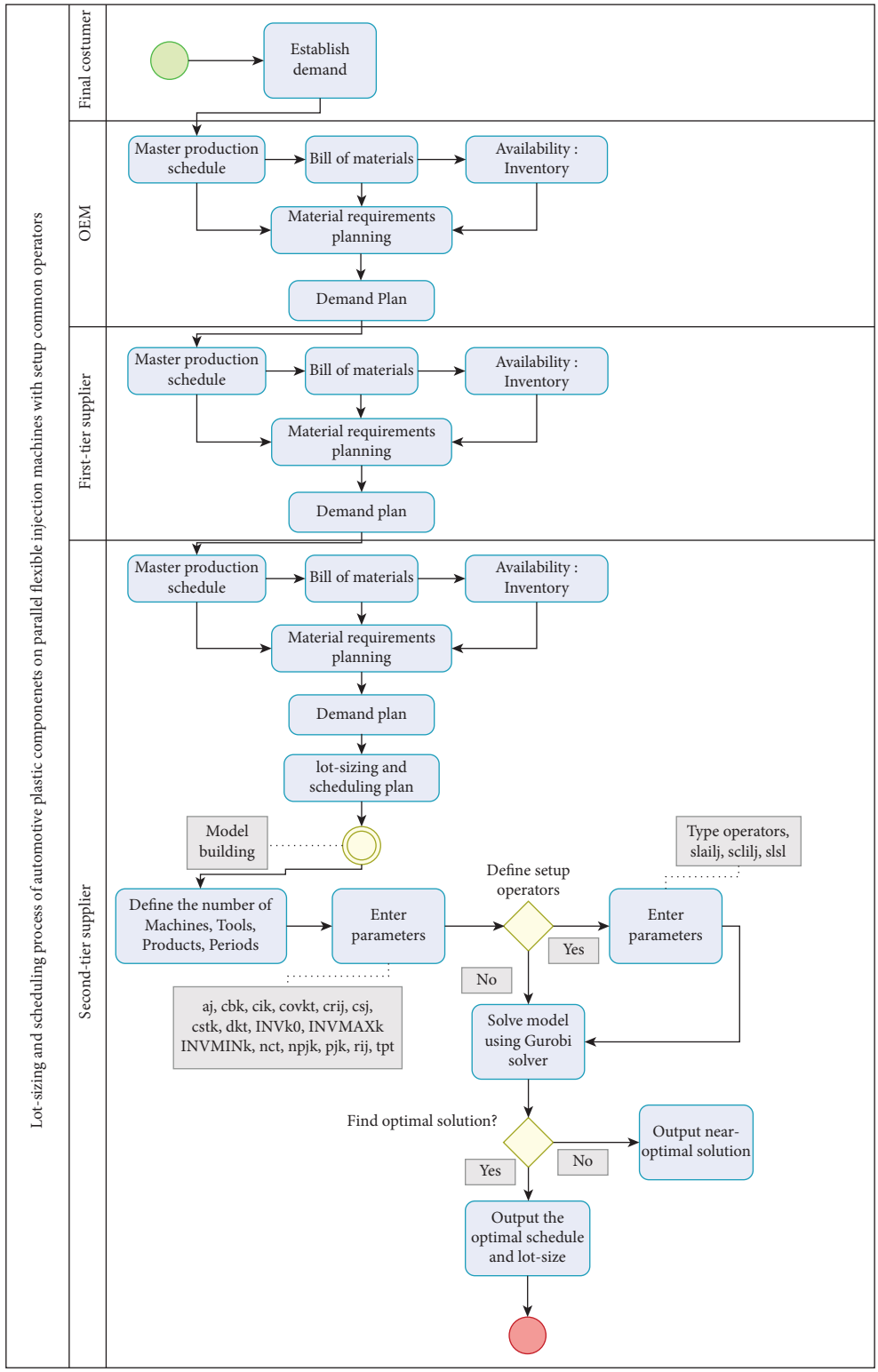


Figure 2: Flowchart of the proposed solution methodology.

Table 2: Nomenclature for the model.

	Index
i	Index of machines $i \in \{1, \ldots, I\}$
j	Index of tools $j \in \{1, \ldots, J\}$
k	Index of products (parts) $k \in \{1, \ldots, K\}$
t	Index of time periods $t \in \{1, \ldots, T\}$
	Data
a_j	Total amount of tools j available for production
cb_k	Backorder cost of product k.
ci_k	Inventory cost of product k
cov_{kt}	Stock coverage defined as the number of time periods for the stock minimum coverage of product k during time period t
cr_{ij}	Setup cost of tool j on machine i
cs_j	Setup cost of preparing tool j
cst_k	Coverage stockout cost of product k.
d_{kt}	Demand of product k during time period t
INV_{k0}	Initial inventory of product k
INVMAX_k	Maximum inventory units for product k during time period t
INVMIN_k	Minimum inventory units for product k during time period t
nc_t	Amount of tool changes allowed during time period t
np_{jk}	Amount of products k no longer produced when tool j is set up
p_{jk}	Amount of products k produced when tool j is set up
r_{ij}	1 if tool j can be set up on machine i, 0 otherwise.
t_{pt}	Production time available during time period t
	Decision variables
B_{kt}	Backorder of product k during time period t
INV_{kt}	Inventory level of product k at the end of time period t
SA_{ijt}	1 if tool j is set up on machine i during time period t and is not set up on machine i during time period $t-1$; 0 if tool j is set up on machine i during time period $t-1$
S_{ijt}	1 if tool j is set up on machine i during time period t, 0 otherwise
ST_{kt}	Coverage stockout of product k during time period t
TP_{ijt}	Production time of tool j set up on machine i during time period t
X_{kt}	Amount of product k to produce during time period t
Xnp_{kt}	Amount of product k no longer produced during time period t, while a tool is set up
X_{kt}'	Amount of product k to produce during time period t by subtracting Xnp_{kt}

subject to sequence constraints:

$$S_{ijt} \leq r_{ij}, \quad \forall i, j, t, \tag{2}$$

$$SA_{ijt} \leq r_{ij}, \quad \forall i, j, t, \tag{3}$$

$$\sum_j S_{ijt} \cdot r_{ij} \leq 1, \quad \forall i, t, \tag{4}$$

$$\sum_i S_{ijt} \cdot r_{ij} \leq a_j, \quad \forall j, t. \tag{5}$$

Constraints (2) and (3) constrain only setup tools j on previously assigned specific machines i. Constraint (4) determines that one or any tool j can be set up for production during each time period t. Constraint (5) guarantees that the total amount of tools j available for production can only be set up as a maximum during each time period t.

Production and capacity constraints:

$$X_{kt} <= \sum_i \sum_j p_{kt} \cdot r_{ij} \cdot TP_{ijt}, \quad \forall k, t, \tag{6}$$

$$Xnp_{kt} <= \sum_j \sum_i np_{jk} \cdot r_{ij} \cdot SA_{ijk}, \quad \forall k, t, \tag{7}$$

$$X_{kt}' = X_{kt} - Xnp_{kt}, \quad \forall k, t, \tag{8}$$

$$TP_{ijt} = tp_t \cdot S_{ijt}, \quad \forall i, j, t. \tag{9}$$

Constraint (6) computes the amount of product k produced during time period t and ensures that a specific tool j is able to be set up on machine i during time period t when product k is produced. Constraint (7) determines the amount of product k no longer produced when tool j is set up during time period t on machine i by considering that another tool j is set up on machine i during time period $t-1$. This also ensures that such a specific tool j is able to be set up on machine i during time period t when product k is produced. Constraint (8) computes the amount of product k to be produced during time period t by subtracting Xnp_{kt} as the amount of product k no longer produced when tool j is set up. Constraint (9) determines the production time used during time period t when tool j is set up on machine i. In TP_{ijt}, we indicate the production lot size during period times, e.g., 24 h. This means that if tool j is set up on machine i, the tool cannot be changed by another one for the next 24 h. Therefore, the minimum lot size corresponds to the products produced during the 24 h that tool j is set up on machine i.

Setup constraints:

$$SA_{ijt} = S_{ijt}, \quad \forall i, j, t = 1, \tag{10}$$

$$SA_{ijt} \geq S_{ijt} - S_{ijt-1}, \quad \forall i, j, t > 1,$$
$$SA_{ijt} \leq 1, \quad \forall i, j, t > 1, \tag{11}$$

$$\sum_{i} \sum_{j} SA_{ijt} \leq nc, \quad \forall t. \tag{12}$$

Constraint (10) allows the first setup of tool j to be determined on machine i, which enables it to be modeled if tool j is set up during time period t on machine i for the first time and decision variables S_{ijt} and SA_{ijt} take the same value 1. Constraint (11) ensures that SA_{ijt} does not take values above 1. Constraint (12) limits the amount of tool changes allowed during time period t.

Inventory balance equations:

$$\mathrm{INV}_{kt} = \mathrm{INV}_{i0} + X'_{kt} - d_{kt} + B_{kt}, \quad \forall k, t = 1, \tag{13a}$$

$$\mathrm{INV}_{kt} = \mathrm{INV}_{kt-1} + X'_{kt} - d_{kt} + B_{kt} - B_{kt-1}, \quad \forall k, t > 1. \tag{13b}$$

Inventory balance equations (13a) and (13b) guarantee appropriate values for the inventories, quantities to produce, and backorders for each time period $t = 1$ and $t > 1$, respectively.

Stock coverage constraint:

$$\mathrm{INV}_{kt} \geq \mathrm{INVMIN}_{kt}, \quad \forall k, t, \tag{14}$$

$$\mathrm{INV}_{kt} \leq \mathrm{INVMAX}_{kt}, \quad \forall k, t, \tag{15}$$

$$\mathrm{INV}_{kt} + ST_{kt} \geq \sum_{c=1}^{c=\mathrm{cov}} \mathrm{d}_{k(t+c)}, \quad \forall k, t < \mathrm{T} - \mathrm{cov}. \tag{16}$$

Constraints (14) and (15) limit the inventory levels for each product k according to the available space for inventory holding during time period t. Constraint (16) is a constraint for the stock coverage of products.

Bound and nature variables:

$$SA_{ijt}, S_{ijt} \in \{0, 1\}, \quad \forall i, j, t, \tag{17}$$

$$X_{kt}, \mathrm{INV}_{kt}, B_{kt}, ST_{kt}, Xnp_{kt}, BX'_{kt}, \in \mathbb{N}, \quad \forall k, t, \tag{18}$$

$$TP_{ijt} \in \mathbb{N}, \quad \forall i, j, t. \tag{19}$$

Constraint (17) indicates the binary nature of setup S_{ijt} and the setup amount SA_{ijt} variables. Constraints (18) and (19) indicate the continuous nature of the represented variables.

5. MILP Model for Lot-Sizing and Scheduling on Parallel Flexible Injection Machines with Setup Common Operators: Notation and Model Formulation

The MILP model for lot-sizing and scheduling on parallel flexible injection machines described in the above section provides a solution to an LSSP that emerged in a second-tier supplier of a real automotive supply chain. In this section, the base model is extended by considering setup common operators to respond and fulfill the constraints that arise in automotive plastic enterprises. In this regard, the base model is extended to offer coming closer to reality by considering another index which represents the workers who change molds. It also bears in mind the casuistry associated with the LSSP problem that refers to setup common operators.

The nomenclature for the extended model is represented in Table 3. In order to avoid repetitions, the data and decision variables used for the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators are considered to be the same as in the base model. Table 3 only shows the new data and decision variables in relation to the previous base model. There are different types of operators and a distinct number of workers corresponding to each operator type. For example, let us consider only one setup operator type that corresponds to the qualified technician category and is specialized in changing molds. Companies have a setup operator type corresponding to the auxiliary technician category, whose task involves helping the qualified technician, as well as a setup operator type corresponding to the mechanic category and provides support whenever failure of a mechanic, electric or physical, among others, occurs. All these categories or operator types have varying numbers of workers who go on different shifts. For example, in a company like that herein studied that has 21 machines, there could be two qualified technicians, four auxiliary technicians, and one mechanical technician per shift.

The formulation of the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators is described next. The objective function minimizes total costs, which include setup costs with common setup operators, inventory costs, penalization costs for coverage stockouts, backorder costs, and tool setup costs, such as the route cost of selecting one machine or another to set up a tool.

$$\mathrm{Min}\, z = \sum_{i}\sum_{l}\sum_{j}\sum_{t} cs_j \cdot SA_{iljt} + \sum_{i}\sum_{l}\sum_{j}\sum_{t} slc_{iljt} \cdot SA_{iljt} + \sum_{k}\sum_{t} ci_k \cdot \mathrm{INV}_{kt} + \sum_{k}\sum_{t} cst_k \cdot ST_{kt} + \sum_{k}\sum_{t} cb_k \cdot B_{it} + \sum_{i}\sum_{l}\sum_{j}\sum_{t} r_{ij} \cdot cr_{ij} \cdot SA_{iljt}, \tag{20}$$

subject to sequence constraints:

Table 3: Nomenclature for the model.

Index	
l	Index setup type operators $l \in \{1, \ldots, L\}$
	Data
sla_{ilj}	Number of setup type operators l required to setup the tool j on machine i
scl_{ilj}	Cost of type operator l to setup the tool j on machine i
sl_{sl}	Number of available workers of type operator l available
	Decision variables
S_{iljt}	1 if the tool j is setup by setup operator l on machine i during time period t, 0 otherwise
SA_{iljt}	1 if tool j is set up by setup operator l on machine i during time period t and is not set up on machine i during time period $t-1$; 0 if tool j is set up by setup operator l on machine i during time period $t-1$

$$S_{iljt} \leq r_{ij}, \quad \forall i, l, j, t, \tag{21}$$

$$SA_{iljt} \leq r_{ij}, \quad \forall i, l, j, t, \tag{22}$$

$$\sum_{j} S_{iljt} \cdot r_{ij} \leq 1, \quad \forall i, l, t, \tag{23}$$

$$\sum_{i} S_{iljt} \cdot r_{ij} \leq a_j, \quad \forall j, t, l. \tag{24}$$

Equations (21) and (22) constrain only setup tools j by setup operators l on specific previously assigned machines m. Constraint (23) determines that one or any tool j can be set up by setup operator l for production during each time period t. Constraint (24) guarantees that the total amount of tools j available can only be set up for production as a maximum during each time period t by setup operator l.

Production and capacity constraints:

$$Xnp_{kt} <= \sum_{i} \sum_{l} \sum_{j} np_{jk} \cdot r_{ij} \cdot SA_{iljk}, \quad \forall k, t, \tag{25}$$

$$TP_{ijt} = tp_t \cdot S_{iljt}, \quad \forall i, l, j, t. \tag{26}$$

Constraint (25) determines the amount of products k no longer produced when tool j is set up by operator l on machine I during time period t by considering that another tool j is set up on machine i during time period $t-1$. It also ensures that such a specific tool j can be set up on machine i during time period t when product k is produced. Constraint (26) determines the production time spent during time period t when tool j is set up by operator l on machine i.

Setup constraints:

$$SA_{iljt} = S_{iljt}, \quad \forall i, l, j, t = 1, \tag{27}$$

$$\begin{aligned} &SA_{iljt} \geq S_{iljt} - S_{iljt-1}, \quad \forall i, l, j, t > 1, \\ &SA_{iljt} \leq 1, \quad \forall i, l, j, t > 1, \end{aligned} \tag{28}$$

$$\sum_{i} \sum_{j} SA_{iljt} \leq nc_t, \quad \forall l, t. \tag{29}$$

Constraint (27) allows the first setup of tool j performed by operator l on machine i to be determined and enables it to be modeled if tool j is set up during time period t on machine i for the first time. Decision variables S_{iljt} and SA_{iljt} take the same value 1. Constraint (28) ensures that SA_{iljt} does not take values above 1. Constraint (29) limits the amount of tool j changes allowed during time period t and set up by operator l on machine i.

Labor constraint:

$$\sum_{i} \sum_{j} SA_{iljt} \cdot sla_{ilj} \leq sls_l, \quad \forall l, t. \tag{30}$$

Constraint (30) limits the amount of tool changes allowed during time period t to the available number of workers of type operator l by considering the number of setup type operators l required to set up tool j on machine i.

Bound and nature variables:

$$SA_{iljt}, S_{iljt} \in \{0, 1\}, \quad \forall i, l, j, t. \tag{31}$$

Constraint (31) indicates the binary nature of the setup S_{iljt} and setup amount SA_{iljt} variables. Finally, the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators is also subject to constraints (6), (8), (13a), (13b), (14)–(16), (18), and (19).

6. Case Study and Computational Experiments

The proposed base MILP model for lot-sizing and scheduling on parallel flexible injection machines and the extended MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators were implemented in Python 3.8.2, using Pyomo [9] as an extensible python-based open-source optimization modeling language for linear programming. The performance of the proposed model was evaluated on a set of instances that reflect different characteristics of the real-world case of the automotive components industry under study. All the numerical tests were conducted on a personal computer equipped with an Intel (R) Core (TM) I5-8500 @ 3.00 GHz Processor and 8 GB RAM. We used Python 3.8.2 and tested applying Gurobi 9.0 to solve the mixed integer linear programming model.

In the next section of data generation, it is described how the data are generated to run the computational experiments. The datasets generated to validate the proposed models correspond to small, medium, and large datasets. Sized datasets can be accessed through a link available at the end of the document. Finally, the last section presents the results of the computational experiments carried out.

6.1. Data Generation. For the data generation, we define various instance sets, including small, medium, and large data sizes. The small dataset corresponds to the minimum amount of data required to test the proposed model; the medium dataset allows to test the model with a reasonable number of parameters and variables to be solved by the model, approaching to the realistic view of the LSSP; and finally, the large dataset replicates the real amount of data managed by real-world enterprises when solving the LSSP. For the computational experiments, all the datasets are built through considering the parameter values depicted in Table 5. The data values are created in the way that mostly represents real data from the automotive components industry; next, the data values are defined as follows:

The parameter a_j determines the total amount of tools j available for production; in this regard, only one unit of each tool is available, and this means that there are not duplicated tools to produce the same components.

The backorder cost (cb_k) and the coverage stockout cost (cst_k) are represented by a very high value (equal to M (99999)) in order to avoid customer missing parts in the model resolution.

Inventory costs (ci_k) are set with values uniformly distributed in given interval; $U(u1, u2)$ is a random variable which is uniformly distributed on $[u_1, u_2]$.

The stock coverage is defined in three days of demand. Nevertheless, in small datasets where the number of periods is lower than three, we have considered one coverage day of demand $(\text{cov}_{kt} = 1)$ in order not to have unfeasible solutions in the model resolution, in such a way that the model considers one period of future demand $(\text{d}t + 1)$ to be produced during period t.

Random $[r_1, r_2]$ values denote a random integer value over the interval from r_1 to r_2. The following data parameters use Random $[r_1, r_2]$ values: setup cost of a tool (cr_{ij}), setup cost of preparing a tool (cs_j), maximum inventory (INVMAXk), and the amount of products no longer produced when a tool is set up (np_{jk}).

The amount of products produced when a tool is set up (p_{jk}) is also denoted as Random $[r_1, r_2]$. In this regard, $[r_1, r_2]$ indicates that product k is assigned to the tool j; otherwise, $p_{jk} = 0$. Parts are randomly assigned to tools, with the condition that each part must be assigned to one tool. Table 4 proposes an example of assignment on a small dataset composed of three tools and six products.

In order to generate the values for the demand, blocks of seven time periods corresponding to the 7 days of the week are considered. In this regard, the first five periods of the week will have demand values set as Random (15, 40); otherwise, $d_{kt} = 0$ on the sixth and seventh periods of the week, that is, on Saturday and Sunday.

The minimum inventory (INVMINk) is set as one unit for all the products k; accordingly, the initial inventory (INV_{k0}) is also set as one unit for all products k.

The amount of tool changes allowed (nct) is defined by Random $(I, I + 5)$ with I being the minimum number of machines changes allowed, which coincides with the total number of machines. Considering the same scheme of blocks of weeks divided in 7 periods, no tool changes are allowed in the 7^{th} period of the week, $nct = 0$; this is because on Sundays the enterprise does not produce, uses it as a day of rest or for machine maintenance, etc.

According to this last statement, the time available for production (tpt)) is 24 hours for the first five periods of the week and 16 hours for the 6^{th} period of the week. No time production is available for the 7^{th} period of the week.

$r_{ij} = 1$ indicates that all the tools can be set up on all the machines.

The aforementioned parameters and values are defined for the base model. The three parameters added for the extended model, which considers setup common operators, are described in Table 5.

$sl_{ailj} = 1$ indicates that one setup type operator is required to setup the tool j on machine i

The cost of type operator l to setup the tool j on machine i (scl_{ilj}) is set with values uniformly distributed in given interval; $U(u_1, u_2)$ is a random variable which is uniformly distributed on $[u_1, u_2]$.

Finally, the number of available workers of type operator l available (sls_l) must be as much as the number of machines I.

In addition, we attach the link to the synthetic data generator, so that the model can be reproduced in future research http://hdl.handle.net/10251/161636.

6.2. Results and Computational Experiments. Using the aforementioned synthetic data generator, a set of experiments have been conducted to validate the two proposed models: (i) base LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines, and (ii) extended LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators.

In order to give the reader a clear insight of the input data parameter values and the output data results once implemented the proposed MILP, we include an example of a small dataset size. The small dataset of the base model is characterized by having 2 machines, 4 tools, 6 parts, and 3 periods. The results obtained with the decision variables in the MILP model for lot-sizing and scheduling on parallel flexible injection machines are presented in Tables 6 and 7.

According to the results obtained in the MILP model for lot-sizing and scheduling on parallel flexible injection machines, all the available capacity is occupied in the three defined periods. Notwithstanding, demand was higher than the available capacity and, therefore, the model had to delay the demand of products $k = 2, 3, 4$. Figure 3 shows the Gantt chart showing the schedule obtained after applying the base

Table 4: Amount of products produced when a tool is set up.

j	k	p_{jk}
1	4	4
2	1	2
2	5	2
2	6	2
3	2	2
4	3	3

Table 5: Generation of values for data parameters.

Parameter	Value
a_j	1
cb_k	99999
ci_k	U (0.1, 1)
cov_{kt}	1 when $T < 3$; otherwise, 3 when $T > 3$
cr_{ij}	Random (5, 15)
cs_j	Random (45, 50)
cst_k	99999
d_{kt}	Random (15, 40) if T = first 5 periods of the week; otherwise, 0 if $T = 6^{th}$ and $T = 7^{\text{th}}$ periods of the week
INV_{k0}	1
INVMAX_k	Random (10000, 20000)
INVMIN_k	1
nc_t	Random $(I, I + 5)$ 0 if $T = 7^{\text{th}}$ period of the week
np_{jk}	Random (2, 5)
p_{jk}	Random (2, 5)
tp_t	24 hours if T = first 5 periods of the week; 16 hours if $T = 6^{\text{th}}$ period of the week; 0 hours if $T = 7^{\text{th}}$ period of the week
r_{ij}	1
sla_{ilj}	1
scl_{ilj}	U (2.5, 3.5)
sls_l	I

Table 6: Small dataset results of the base LSSP MILP: lot-sizing, inventories, coverage stockout, and backorders.

k	t	X_{kt}	Xnp_{kt}	X'_{kt}	ST_{kt}	INV_{kt}	B_{kt}
1	1	48	0	48	2	21	0
1	2	27	0	27	0	25	0
1	3	0	0	0	0	1	0
2	1	0	0	0	33	1	35
2	2	48	0	48	30	1	21
2	3	48	0	48	0	1	4
3	1	72	0	72	0	39	0
3	2	0	0	0	11	19	0
3	3	0	0	0	0	1	12
4	1	0	0	0	14	1	30
4	2	0	0	0	32	1	45
4	3	78	0	78	0	1	0
5	1	48	0	48	8	31	0
5	2	41	0	41	0	33	0
5	3	0	0	0	0	1	0
6	1	48	0	48	19	15	0
6	2	48	0	48	5	29	0
6	3	0	0	0	0	1	6

MILP model. Each row represents a machine i and each rectangle a tool j; inside the rectangle, we have indicated the parts k and processing time t of each tool j.

Table 7: Small dataset results of the base LSSP MILP: scheduling.

i	j	t	SA_{ijt}	S_{ijt}	TP_{ijt}
1	2	1	1	1	24
1	2	2	0	1	24
1	1	3	1	1	24
2	4	1	1	1	24
2	3	2	1	1	24
2	3	3	0	1	24

The input and output data for the small, medium, and large datasets of the base LSSP model are presented at https://doi.org/10.4995/Dataset/10251/161642.

The input data of the extended LSSP model are characterized by considering 2 machines, 4 tools, 6 parts, 3 periods, and 4 type operators. The results obtained with the decision variables in the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators are presented in Tables 8 and 9.

According to the obtained results, all the available capacity is occupied along the three defined periods. Nevertheless, demand was higher than the available capacity and, therefore, the model had to delay the demand of products $k = 2, 4$ in periods $t = 1, 2$ and had to delay the demand of products $k = 3, 6$ in period $t = 3$. The solution provided in

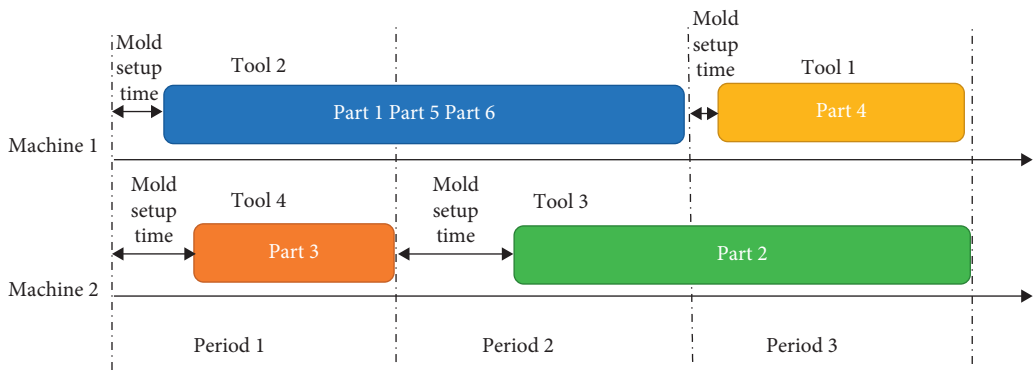


Figure 3: Gantt chart base MILP model for lot-sizing and scheduling on parallel flexible injection machines.

Table 8: Small dataset results of the extended LSSP MILP, with setup common operators: lot-sizing, inventories, coverage stockout, and backorders.

k	t	X_{kt}	Xnp_{kt}	X'_{kt}	ST_{kt}	INV_{kt}	B_{kt}
1	1	48	0	48	2	21	0
1	2	27	0	27	0	25	0
1	3	0	0	0	0	1	0
2	1	0	0	0	33	1	35
2	2	48	0	48	30	1	21
2	3	48	0	48	0	1	4
3	1	72	0	72	0	39	0
3	2	0	0	0	11	19	0
3	3	0	0	0	0	1	12
4	1	0	0	0	14	1	30
4	2	0	0	0	32	1	45
4	3	78	0	78	0	1	0
5	1	48	0	48	8	31	0
5	2	41	0	41	0	33	0
5	3	0	0	0	0	1	0
6	1	48	0	48	19	15	0
6	2	48	0	48	5	29	0
6	3	0	0	0	0	1	6

Table 9: Small dataset results of the base LSSP MILP: scheduling.

i	l	j	t	S_{iljt}	SA_{iljt}	TP_{ijt}
1	1	2	1	1	1	24
1	2	2	1	1	1	0
2	1	4	1	1	1	0
2	2	4	1	1	1	0
1	2	4	1	0	0	24
2	1	3	2	1	1	0
2	2	3	2	1	1	0
1	1	2	2	1	0	24
1	2	2	2	1	0	0
1	2	3	2	0	0	24
1	1	1	3	1	1	24
1	2	1	3	1	1	0
2	1	3	3	1	0	0
2	2	3	3	1	0	0
1	2	3	3	0	0	24

Table 10: Experimental results for the MILP model for lot-sizing and scheduling on parallel flexible injection machines.

Dataset	I	J	K	T	Number of constraints	Number of variables	Number of binary variables	Number of integer variables	Number of continuous variables	Number of nonzeros	Objective value	GAP (%)	Termination condition	Time (sec)
S1	2	4	6	3	253	180	48	48	84	548	30700049.77	0.00	Optimal	0.1875
S2	4	6	8	3	529	360	144	144	72	1240	13400404.31	0.00	Optimal	0.0312
S3	6	8	16	3	1037	720	288	288	144	2672	11100863.25	0.00	Optimal	0.0156
S4	8	10	22	3	1617	1116	480	480	156	4372	8101150.62	0.00	Optimal	0.0781
M1	10	12	24	14	10882	7056	3360	3360	336	29232	75409783.76	0.00	Optimal	2.9056
M2	12	14	28	14	14630	9408	4704	4704	0	39872	162208156.70	0.00	Optimal	9.8883
M3	14	16	32	14	18930	12096	6272	6272	-448	52160	129012740.08	0.00	Optimal	19.5891
M4	16	18	36	14	23782	15120	8064	8064	-1008	66096	109311635.70	0.00	Optimal	24.0256
L1	18	20	40	14	29186	18480	10080	10080	-1680	81680	227015024.17	0.00	Optimal	2.0308
L2	20	40	60	14	61754	38640	22400	22400	-6160	164880	698925080.54	0.20	maxTimeLimit	18000.6893
L3	25	45	70	14	85269	53130	31500	31500	-9870	230910	542429122.73	0.00	Optimal	3786.1561
L4	30	50	80	14	112234	69720	42000	42000	-14280	307240	834529827.82	0.51	maxTimeLimit	18001.5215

Table 11: Experimental results for the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators.

Dataset	I	J	K	L	T	Number of constraints	Number of variables	Number of binary variables	Number of integer variables	Number of continuous variables	Number of nonzeros	Objective value	GAP (%)	Termination condition	Time (sec)
S1	2	4	6	2	3	392	228	96	96	36	880	30700299.29	0.00	Optimal	0.015592575
S2	4	6	8	2	3	904	504	288	288	-72	2224	9700768.824	0.00	Optimal	0.015617609
S3	6	8	16	2	3	1760	1008	576	576	-144	4736	23501097.9	0.00	Optimal	0.062484741
S4	8	10	22	2	3	2800	1596	960	960	-324	7860	18901770.98	0.00	Optimal	0.031243086
M1	10	12	24	4	14	36744	17136	13440	13440	-9744	100032	67810309.09	0.00	Optimal	12.21586823
M2	12	14	28	4	14	50596	23520	18816	18816	-14112	138992	99214112.49	0.00	Optimal	26.69681978
M3	14	16	32	4	14	66656	30912	25088	25088	-19264	184320	143015128.2	0.00	Optimal	14.35598087
M4	16	18	36	4	14	84924	39312	32256	32256	-25200	236016	146316474	0.00	Optimal	79.8799355
L1	18	20	40	6	14	156200	68880	60480	60480	-52080	432320	211521532.7	0.00	Optimal	241.9122446
L2	20	40	60	8	14	454244	195440	179200	179200	-162960	1212080	867861481	1.54	maxTimeLimit	18015.20472
L3	25	45	70	12	14	950246	399630	378000	378000	-356370	2545660	915900800.2	5.39	maxTimeLimit	18021.99296
L4	30	60	120	15	14	1893720	791280	756000	756000	-720720	5256960	1359169510	4.31	maxTimeLimit	18065.29413

the scheduling shows that the two type operators l are assigned to change the tools j.

The experimental results obtained with the tests conducted for all the datasets employed in the validation for both the base model and the extended model in which the labor index is added are presented in Tables 10 and 11. The input and output data for the small, medium, and large datasets of the extended LSSP model are presented at https://doi.org/10.4995/Dataset/10251/161643. Four instances are generated of each dataset size. The synthetic instances for the base model have been generated through the aforementioned proposed generator: http://hdl.handle.net/10251/161636. The synthetic instances for the extended model with common setup operators have been generated through the following generator: http://hdl.handle.net/10251/161635.

The computational results (Table 10) show that the MILP model for lot-sizing and scheduling on parallel flexible injection machines in small (S1, S2, S3, and S4) and medium (M1, M2, M3, and M4) instances achieves optimal results (GAP = 0%) in, at maximum, 24 seconds. With regard to the large instances generated (L1, L2, L3, and L4), the calculation time on average is set as 2.8 hours, finding a very near-optimal solution, with an average 0.24% GAP. In this regard, it is worth to highlight that the large dataset L1 achieves the optimal solution in a very reduced computational time of 2 seconds.

The computational results (Table 11) show that the MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators in small (S1, S2, S3, and S4) and medium (M1, M2, M3, and M4) instances achieves optimal results (GAP = 0%) in, at maximum, 79.9 seconds. With regard to the large instances generated (L1, L2, L3, and L4), the calculation time on average is set as 3.7 hours finding a very near-optimal solution, with an average 2.81% GAP. In this regard, it is worth to highlight that the large dataset L1 achieves the optimal solution in a very reduced computational time of 242 seconds.

The experiments carried out for both the base model and the extended model that consider common setup operators are valid for its application on solving LSSP models with real amount of data managed by real-world enterprises.

7. Discussion and Conclusions

This paper addresses the LSSP applied to an automotive plastic components' enterprise. An MILP base model is proposed to deal with the lot-sizing and scheduling problem on parallel flexible injection machines to mainly minimize the setup, inventory, stockout, and backorder costs by taking into account injection molds as the main index to schedule parallel flexible injection machines. The MILP base model is extended to provide the enterprise under study with a more realistic solution that considers setup common operators. Therefore, the extended LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators, is presented. This produces a model that adapts to the restrictions of the company under study, an automotive plastic components' enterprise. The peculiarity of this model is that it takes injection molds as the main index to schedule parallel flexible injection machines by considering setup common operators. The novelty of both proposed models lies in our study considering many periods when modeling and running experiments to solve the LSSP. Moreover, the proposed MILP bears in mind stock coverage constraints, which are typical in the studied automotive supply chain industry context, and contemplates an objective function that allows idle times among molds, which is a fundamental characteristic for real cases and has been ignored by former studies.

Finally, this paper validates the proposed MILP by performing experiments with different sized instances, including small, medium, and large datasets. The large dataset is characterized by replicating the amount of data used in the real enterprise that is the object of this study. The goodness of the model is evaluated with the computational time and the deviation of the obtained results as regards the optimal solution.

This study is not without its limitations. The small and medium datasets are solved in both cases in very efficient computing times. The application of the proposed model using the large dataset is more limited in computational efficiency terms. To solve this problem, the literature indicates the generation of heuristics, metaheuristics, and matheuristics. The last type falls within the authors' future research options as far as the contemplated model herein is concerned. Thus, the first research line intends to improve the computational efficiency to solve the model by applying matheuristics, which would consist in solving the binary variables in both the base model (S_{ijt} and SA_{ijt}) and its extended version (S_{iljt} and SA_{iljt}) by a metaheuristic technique, e.g., genetic algorithms, taboo search, and simulated annealing. The metaheuristics result will be provided as input data for the MILP. The following research lines focus on improving the model by considering new constraints that will be very useful for the real firm. In this way, the second future research line is a second extension of the base model that bears in mind the availability of materials. For this purpose, inventory equations are to be added that consider the list of materials needed to manufacture end products. Finally, for the third future research line, a third extension of the model is proposed by considering space limitations in a warehouse's volume. In this way, part volumes are calculated to meet both stock coverage and limited warehouse space, contemplated from the premise that all parts have different volume requirements. This means that the model and its third extension (or third generation) will be capable of meeting the stock coverage of large-sized parts and stocking small-sized ones so that when they have to be produced, the production resources related to bigger pieces will be used. The third-generation model will allow bigger-sized parts to be stored, which will be left at the stock coverage level by calculating the quantity of smaller parts whose coverage can be extended.

8. Data Availability

The instances generated (input and output datasets) and analyzed during the study are available at the following links: (i) base LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines: https://doi.org/10.4995/Dataset/10251/161642;

(ii) extended LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators: https://doi.org/10.4995/Dataset/10251/161643. The algorithms developed for generating the synthetic datasets are available at the following links: (i) base LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines: http://hdl.handle.net/10251/161636; (ii) extended LSSP model, an MILP model for lot-sizing and scheduling on parallel flexible injection machines with setup common operators: http://hdl.handle.net/10251/161635.

9. Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

10. References

[1] R. Gujjula, S. Werk, and H.-O. Günther, A heuristic based on Vogel's approximation method for sequencing mixed-model assembly lines," International Journal of Production Research, vol. 49, no. 21, pp. 6451–6468, 2011.10.1080/00207543.2010.527384

[2] S.-i. Kim, J. Han, Y. Lee, and E. Park, "Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum," Computers & Operations Research, vol. 37, no. 2, pp. 302–314, 2010.10.1016/j.cor.2009.05.007

[3] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," Business & Information Systems Engineering, vol. 6, no. 4, pp. 239–242, 2014.10.1007/s12599-014-0334-4

[4] S. Echchakoui and N. Barka, "Industry 4.0 and its impact in plastics industry: a literature review," Journal of Industrial Information Integration, vol. 20, no. p. 100172, 2020.10.1016/j.jii.2020.100172

[5] H2020 Project C2NET, "'Cloud Collaborative Manufacturing Networks' (C2NET)." 2017.

[6] R. Sanchís, B. Andrés, R. Poler, J. Mula, and M. Díaz-Madroñero, "La solución de optimización C2NET," Dirección y Organización, vol. 64, 64, pp. 36–41, 2018.10.37610/dyo.v0i64.520

[7] K. P. Abdul Nazar and V. Madhusudanan Pillai, "Mixed-model sequencing problem under capacity and machine idle time constraints in JIT production systems," Computers & Industrial Engineering, vol. 118, pp. 226–236, 2018.10.1016/j.cie.2018.02.032

[8] J. Mula, A. C. Lyons, J. E. Hernández, and R. Poler, "An integer linear programming model to support customer-driven material planning in synchronised, multi-tier supply chains," International Journal of Production Research, vol. 52, no. 14, pp. 4267–4278, Jan. 2014.10.1080/00207543.2013.878055

[9] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, Pyomo - Optimization Modeling in Python, 1st ed. Springer Publishing Company, Berlin, Germany, 2012

[10] Y. Á. Ríos-Solís, O. J. Ibarra-Rojas, M. Cabo, and E. Possani, "A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production," European Journal of Operational Research, vol. 284, no. 3, pp. 861–873, 2020.10.1016/j.ejor.2020.01.016

[11] O. J. Ibarra-Rojas, R. Z. Ríos-Mercado, Y. A. Rios-Solis, and M. A. Saucedo-Espinosa, "A decomposition approach for the piece-mold-machine manufacturing problem," International Journal of Production Economics, vol. 134, no. 1, pp. 255–261, 2011.10.1016/j.ijpe.2011.07.006

[12] Y. Li, Z. Yang, G. Li, D. Zhao, and W. Tian, "Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties," IEEE Transactions on Industrial Electronics, vol. 66, no. 2, pp. 1565–1575, Feb. 2019.10.1109/tie.2018.2840498

[13] L. Guimarães, D. Klabjan, and B. Almada-Lobo, "Modeling lotsizing and scheduling problems with sequence dependent setups," European Journal of Operational Research, vol. 239, no. 3, pp. 644–662, 2014.10.1016/j.ejor.2014.05.018

[14] H. Stadtler, "Multi-level single machine lot-sizing and scheduling with zero lead times," European Journal of Operational Research, vol. 209, no. 3, pp. 241–252, 2011.10.1016/j.ejor.2010.09.022

[15] C. Wolosewicz, S. Dauzère-Pérès, and R. Aggoune, "A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem," European Journal of Operational Research, vol. 244, no. 1, pp. 3–12, 2015.10.1016/j.ejor.2015.01.034

[16] R. J. W. James and B. Almada-Lobo, "Single and parallel machine capacitated lotsizing and scheduling: new iterative MIP-based neighborhood search heuristics," Computers & Operations Research, vol. 38, no. 12, pp. 1816–1825, 2011.10.1016/j.cor.2011.02.005

[17] H. Meyr and M. Mann, "A decomposition approach for the general lotsizing and scheduling problem for parallel production lines," European Journal of Operational Research, vol. 229, no. 3, pp. 718–731, 2013.10.1016/j.ejor.2013.03.036

[18] M. Díaz-Madroñero, J. Mula, B. Andres, R. Poler, and R. Sanchis, "A capacitated lot-sizing and scheduling model for the bi-part injection moulding problem," In Proceeding of the 11th International Conference on Industrial Engineering and Industrial Management (CIO2017),Valencia, Spain July 2017.

11. Acknowledgments

This work was supported by the Conselleria de Educación, Investigación, Cultura y Deporte-Generalitat Valenciana for hiring predoctoral research staff with Grant no. ACIF/2018/170

and European Social Fund with Grant Operational Program of FSE 2014–2020, the Valencian Community, and the authors would like to acknowledge the support of the researchers participating in the collaborative projects "Cloud Collaborative Manufacturing Networks" (C2NET) (http://c2net-project.eu/), which has received funding from the EU Horizon 2020 Research and Innovation Programme with grant agreement no. 63690, and "Zero Defects Manufacturing Platform" (ZDMP) (http://www.zdmp.eu), which has received funding from the EU Horizon 2020 Research and Innovation Programme with grant agreement no. 825631.

References

[1] B. Andres, E. Guzman, and R. Poler, "A novel MILP model for the production, lot sizing, and scheduling of automotive plastic components on parallel flexible injection machines with setup common operators," *Complexity*, vol. 2021, Article ID 6667516, 16 pages, 2021.

Hindawi
Complexity
Volume 2021, Article ID 8870356, 15 pages
https://doi.org/10.1155/2021/8870356

Research Article

An Adaptive Reference Vector Adjustment Strategy and Improved Angle-Penalized Value Method for RVEA

Wenbo Qiu,[1] Jianghan Zhu,[1] Huangchao Yu,[2] Mingfeng Fan,[3] and Lisu Huo[1]

[1]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China*
[2]*Institute of Unmanned System, National University of Defense Technology, Changsha 410073, Hunan, China*
[3]*School of Traffic and Transportation Engineering, Central South University, Changsha 410073, Hunan, China*

Correspondence should be addressed to Huangchao Yu; yuhc1221@nudt.edu.cn

Received 16 September 2020; Revised 6 October 2020; Accepted 21 October 2021; Published 17 November 2021

Academic Editor: Inés P. Mariño

Decomposition-based evolutionary multiobjective algorithms (MOEAs) divide a multiobjective problem into several subproblems by using a set of predefined uniformly distributed reference vectors and can achieve good overall performance especially in maintaining population diversity. However, they encounter huge difficulties in addressing problems with irregular Pareto fronts (PFs) since many reference vectors do not work during the searching process. To cope with this problem, this paper aims to improve an existing decomposition-based algorithm called reference vector-guided evolutionary algorithm (RVEA) by designing an adaptive reference vector adjustment strategy. By adding the strategy, the predefined reference vectors will be adjusted according to the distribution of promising solutions with good overall performance and the subspaces in which the PF lies may be further divided to contribute more to the searching process. Besides, the selection pressure with respect to convergence performance posed by RVEA is mainly from the length of normalized objective vectors and the metric is poor in evaluating the convergence performance of a solution with the increase of objective size. Motivated by that, an improved angle-penalized distance (APD) method is developed to better distinguish solutions with sound convergence performance in each subspace. To investigate the performance of the proposed algorithm, extensive experiments are conducted to compare it with 5 state-of-the-art decomposition-based algorithms on 3-, 5-, 8-, and 10-objective MaF1–MaF9. The results demonstrate that the proposed algorithm obtains the best overall performance.

1. Introduction

In the real world, decision makers often encounter some problems with more than one objective to be solved simultaneously. These problems are called multiobjective optimization problems (MOPs), and if the number of objectives is larger than 3, they are termed as many-objective optimization problems (MaOPs). An MOP or MaOP can be formulated as follows:

$$\begin{aligned} &\min F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})], \\ &\text{s.t.}\mathbf{x} \in \Omega, \end{aligned} \tag{1}$$

where $\Omega \subset R^n$ denotes the decision space and F is a map from R^n to R^M, i.e., from the decision space to the objective space. $f_i(\mathbf{x})$ is the ith objective value of the problem and M is the number of objectives. Mostly, the objectives of an MOP or MaOP are contradictory to each other and thus there rarely exists a solution that can surpass all the other solutions on each objective. As a result, a set of solutions representing the trade-off among all the objectives, called Pareto-optimal solutions, can be achieved and pursued by researches. The Pareto-optimal solutions are known as Pareto front (PF) in the objective space and Pareto set (PS) in the decision space, respectively.

To address the multiobjective problems (MOPs) and many-objective problems (MaOPs) appearing in many real-world industries, e.g., data mining [1], airline crew roster recovery [2], and software engineering [3], multiobjective evolutionary algorithms (MOEAs) and many-objective evolutionary algorithms (MaOEAs) are seen as the major methods for approximating PF of MOPs because of its

population-based nature. During the past decades, a large number of algorithms have been developed such as Nondominated Sorting Genetic Algorithm II (NSGA-II) [4], multiobjective evolutionary algorithm based on decomposition (MOEA/D) [5], and so on [6]. According to their environmental selection strategies, the existing algorithms can be roughly divided into four categories: (1) Pareto-dominance-based; (2) decomposition-based; (3) indicator-based; and (4) others. Pareto-dominance-based algorithms [4, 7, 8] often divide solutions into different nondominated levels and use a second criterion to select solutions in the last level; decomposition-based MOEAs [9–13] decompose the original MOP into multiple subproblems and solve them in a cooperative way; for indicator-based MOEAs and MaOEAs, such as hypervolume-based many-objective (HypE) [14] and indicator-based multiobjective evolutionary algorithm with reference point adaptation (AR-MOEA) [15], it tends to develop an indicator to evaluate the overall performance and sort the individuals according to their indicator values.

Among the four categories, a branch of decomposition-based MOEAs and MaOEAs divides the whole objective space into a number of subspaces by a set of predefined uniformly distributed reference vectors or reference weights, such as RVEA [16] and MOEA/D-M2M [17]. This branch shows promising performance in maintaining population diversity. However, there are still some shortcomings in this branch of algorithms. Firstly, when the MOP owns irregular PF, e.g., degenerated or disconnected, the PF is not uniformly distributed in the objective space, and thus the predefined reference vectors are not able to evenly divide the PF which may affect the final quality of the output population. The PFs of 3-objective DTLZ7 and MaF6 are shown in Figure 1 where the PF of DTLZ7 is disconnected and that of MaF6 is degenerated.

Besides, although the PFs of the above two problems are irregular, the PF of 2 or 3 objectives can still be divided to some degree of satisfaction because the objective space is not large and the predefined reference vectors can still be densely distributed in the area where the PF is located. However, with the increase of the number of objectives, the objective space expands sharply and the distribution of reference vectors in the objective space is very sparse. As a result, sometimes the PF is divided by few reference vectors. To visually describe this phenomenon, an example about the optimization of 10-objective MaF1 by RVEA is shown in Figure 2.

In RVEA, a reference vector is called active when at least one candidate solution is associated with it. Figure 2 shows the proportion of active reference vectors among all the predefined reference vectors during the searching process, and the population size is set as 275. We can see that only less than 10% of the predefined reference vectors are working.

To cope with the above issues, a new decomposition-based evolutionary algorithm is proposed in this paper. In the newly proposed algorithm, a new definition termed as angle position is developed to evaluate the position of a candidate solution in the objective space (the new algorithm is thus called AP-RVEA). Then, the reference vectors are adaptively adjusted according to the distribution of the candidate solutions on the basis of the definition of angle position. The inactive reference vectors will be deleted and some candidate solutions will be constructed into new reference vectors according to their positions relative to the active reference vectors in the objective space. In this way, the subspaces in which the true PF lies will be further divided.

Besides, the metric in APD value (the selection indicator used in RVEA) responsible for evaluating the convergence of solutions is the length of the normalized objective vector. This metric is poor at posing enough selection pressure in terms of convergence performance with the increase of the objective size. To fix the above issue, an improved APD value is proposed in AP-RVEA to keep promising solutions in problems with many objectives.

The main contributions of this paper can be summarized as follows:

(1) A novel adaptive reference vector adjustment strategy based on the positions of candidate solutions in the objective space relative to the current active reference vectors is proposed. With the strategy, the predefined reference vectors will be adjusted according to the distribution of promising solutions, which can be considered as an approximation of the true PF in some way. During each generation that runs the strategy, the subspaces in which the true PF lies may be further divided into more subspaces and contribute more to that final result.

(2) To cope with the poor selection pressure in terms of convergence performance posed by APD value, an improved APD value method is designed to better evaluate the convergence performance of solutions in each subspace.

The rest of this paper is organized as follows. Section 2 gives a survey of strategies about reference vector adjustment. Section 3 introduces some basic definitions of the proposed AP-RVEA. In Section 4, the details of the proposed AP-RVEA are described. Section 5 presents the experimental results of AP-RVEA and other 5 algorithms on MaF [18] test suite. Finally, the conclusion and future work are covered in Section 6.

2. Related Studies

Reference vector-based MOEAs or MaOEAs are efficient in maintaining population diversity when dealing with MOPs or MaOPs [19, 20]. However, since the shape of PF is unknown in advance for most MaOPs, using predefined reference vectors to acquire a set of evenly distributed solutions might not be feasible especially when the problem has an irregular PF. As a result, some research studies about the adjustment of reference vectors have been developed in recent years.

Jiang et al. [21] proposed a asymmetric Pareto-adaptive (apa) scheme for MOEA/D [5] to deal with problem with symmetric and asymmetric Pareto fronts. The apa scheme was useful when the PF of the problem satisfied that $f_1^p + f_2^p + \cdots + f_M^p = 1$, where p is a parameter to estimate

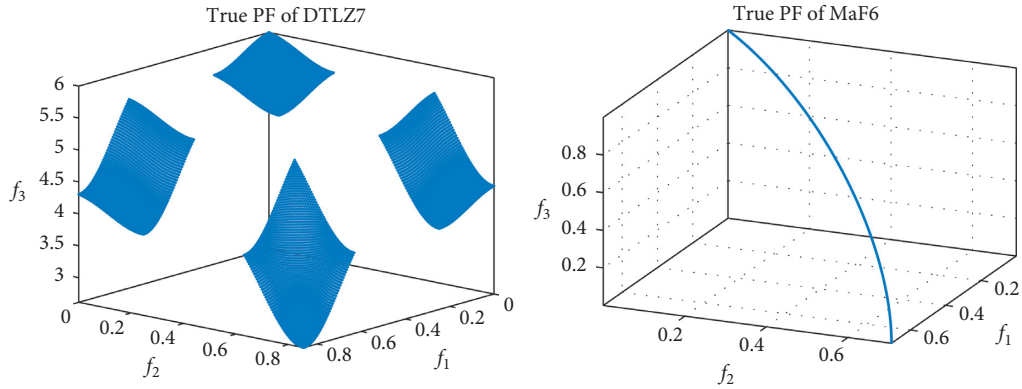


Figure 1: The true PFs of DTLZ7 and MaF6.

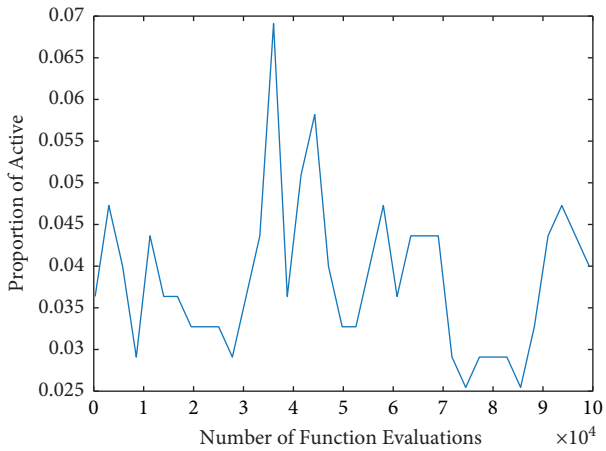


Figure 2: The proportion of active reference vectors during the optimization of 10-objective MaF1 by RVEA.

the shape of the PF and M is the number of objectives. In EMOSA [22], a set of Q evenly distributed weight vectors (reference vectors) and a set of C_{H+M-1}^{m-1} predefined candidate wight vectors P^* are maintained during the optimization process, where H denotes a positive integer and $C_{H+M-1}^{m-1} \gg Q$. For a solution x_1 (associated with λ_1), its closest nondominated neighbor x_2 (associated with λ_2) is labeled. The new weight vector λ is chosen from P^* and λ_1 is replaced if and only if (1) $\text{dist}(\lambda_1, \lambda_2) < \text{dist}(\lambda, \lambda_2)$) and (2) $\text{dist}(\lambda, \lambda_1) \leq \text{dist}(\lambda, P^*)$.

In RVEA* [16], for each inactive subspace (empty subspace), the corresponding reference vector would be deleted and replaced by a unit vector that was randomly generated inside the range of the minimum and maximum value. Wang et al. [23] also proposed an algorithm called PICEA-w where weights were co-evolved with candidate solutions during the optimization process. As a result, suitable weights could be adaptively constructed and guide the candidate solution to the true Pareto front (PF) effectively.

In Qi et al. [24], an improved MOEA/D with adaptive weight vector adjustment was proposed to handle problems with irregular PFs. During the searching process, the weights were periodically adjusted to redistribute the weights of the subproblems so as to obtain better uniformity of solutions. In Jiang [25], new reference vectors were chosen from the combination of parent population and offspring according to their distances to the current reference vectors. In the first step of the algorithm, all the extreme solutions would be selected into the empty reference vector set.

Liang et al. [26] developed two reference vector adaptation strategies called scaling of reference vectors (SRV) and transformation of solutions location (TSL) for many-objective evolutionary algorithms. SRV introduced a center vector and adjusted the other reference vectors around by a scaling function; TSL transformed the promising solutions in the current generation into a set of new reference vectors. Cai et al. [27] also proposed a decomposition-based MaOEA with two types of adjustments for direction vectors where a Pareto-dominance-based mechanism was used to evaluate the effectiveness of each direction vector and the ineffective direction vectors would be adjusted to better fit the shape of the true PF.

Asafuddoula et al. [28] proposed an enhanced decomposition-based evolutionary algorithm with adaptive reference vectors called G-DBEA. In G-DBEA, two types of reference vectors called active and inactive reference vectors were maintained. During the search process, if an offspring preferred an inactive reference vector, the vector would be removed from the set of inactive reference vectors and the corresponding reference vector of the offspring would be deleted from the set of active reference vectors. At the same time, if a reference vector was not associated with any nondominated solutions in a certain period, it will be deleted from the set of active reference vectors and the solution with the smallest perpendicular distance to the reference vector would be constructed as a new reference vector.

Zhao et al. [29] proposed a modified decomposition-based many-objective ant colony optimization (ACO) algorithm and employed an adaptive reference point mechanism that chooses the ideal point or nadir point as the reference point according to the distribution of the candidate solutions. Zhou et al. [30] developed a novel entropy-based evolutionary algorithm with adaptive reference points called EARPEA where entropy computed based on reference points and a learning period are employed to control adaptation of the reference points. Wang et al. [31] gave a new

preference-based MOEAs called MOEA/D-AWV in which the weight vectors are generated adaptively by the decision maker's preference and finally guide the solutions to converge to a preferred distribution.

In Deb et al. [32], reference points with no associated solution would be directly deleted and a simplex of M reference points around a remaining reference point would be added. In Cheng et al. [33], a solution that had the least similarity from active reference vectors was found and normalized, and then an inactive reference vector would be randomly deleted and replaced by the normalized solution. Liu et al. [34] used the growing neural gas network to learn the distribution of the reference vectors, thus achieving automatic yet stable adaptation.

In the proposed AP-RVEA, inactive reference vectors will be deleted after a period of generations. Besides, the newly defined angle position works to evaluate the position of solutions with promising performance against the reference vectors. Then, new reference vectors will be chosen on the basis of the information given by angle position. The core idea of the AP-RVEA is to adaptively adjust the reference vectors to further divide the subspaces where the true PF may lie. Furthermore, an improved version of APD method called F-APD based on the fractional dominance relation is developed to distinguish the convergence performance of the solutions. They cooperate with each other to achieve promising convergence and diversity and sound balance between them.

3. Background

3.1. The Objective Space Decomposition Strategy. In this section, an objective space decomposition strategy is shown to divide the objective space into a set of subspaces [17]. Each subspace owns a set of solutions and can be seen as a subproblem. In other words, a population is divided into a set of subpopulations to maintain diversity for the whole population. In each generation, all the subpopulations are optimized together.

Definition 1 (subspace). At first, N uniformly distributed unit vectors: $u^1, u^2, \ldots, u^N$, are predefined to divide the objective space into N subspaces $\Omega^1, \Omega^2, \ldots, \Omega^N$. Then, each solution will be associated with an subspace according to their acute angles between the solution and unit vectors. $\Omega^i = \{v | \langle v, u^i \rangle \leq \langle v, u^j \rangle, \forall j \in 1, 2, \ldots, N\}$, where $\langle v, u^i \rangle$ denotes the acute angle between v and u^i. According to the definition of subspace, a vector v belongs to Ω^i, if and only if v has the smallest angle to u^i compared to other unit vectors. Each solution is to be associated with an unit vector and in a subspace during the optimization process. u^i is the corresponding reference vector of all the solutions in Ω^i and the acute angle of v to Ω^i is the acute angle of v to u^i.

Definition 2 (neighborhood subspace). Let K be the size of neighborhood spaces. For subspace Ω^i ($i = \{1, 2, \ldots, N\}$), its neighborhood subspaces are defined as $\mathrm{NS}_i = \{\Omega^{i_1}, \ldots, \Omega^{i_K}\}$, where $\forall \Omega^j \in \mathrm{NS}_i, \forall \Omega^k \notin \mathrm{NS}_i, \langle u^j, u^i \rangle \leq \langle u^k, u^i \rangle$. In other words, the neighborhood subspaces of subspace Ω^i are the union of subspaces whose unit vectors have the first K smallest acute angles to u^i among all the unit vectors.

3.2. RVEA. On the basis of the above objective space decomposition strategy, RVEA [16] aims to maintain a sound balance between convergence and diversity for the population. It adopts the acute angles between solutions and the reference vectors and the length of the objective vectors to assess the overall performance of solutions by using the following formula called angle-penalized distance (APD):

$$d_{t,i,j} = \left(1 + P\left(\theta_{t,i,j}\right)\right) \cdot \left\| f'_{t,i} \right\|, \tag{2}$$

where $P(\theta_{t,i,j})$ and $\|f'_{t,i}\|$ measure the diversity performance and convergence performance, respectively, $\theta_{t,i,j}$ is the acute angle between the ith objective vector and u^i, and $P(\theta_{t,i,j})$ is a penalty function related to $\theta_{t,i,j}$:

$$P\left(\theta_{t,i,j}\right) = M \cdot \left(\frac{t}{t_{\max}}\right)^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{u_{t,j}}},$$
$$\gamma_{u_{t,j}} = \min \langle u^i, u^j \rangle, \quad i \in 1, 2, \ldots, N, i \neq j, \tag{3}$$

where N denotes the population size and M is the objective number. $\gamma_{u_{t,j}}$ is the minimum angle between u^i and other reference vectors. The value of α controls the changing rate of $P(\theta_{t,i,j})$ and a larger α means that more emphasis will be allocated to the convergence performance compared to the diversity performance.

In the environmental selection of RVEA, a solution will be selected from each subspace, and among all the solutions in a subspace, the solution being assigned with the minimum APD method will be selected.

3.3. The Definition of Angle Position. In this paper, we try to further divide the active subspaces during the optimization of algorithms with the objective space decomposition strategy. Consequently, it is essential to know the accurate position of a solution to the active reference vectors. For this purpose, the definition of angle position is proposed.

Definition 3 (neighborhood reference vector). At first, the definition of neighborhood reference vectors of a solution is defined. Let M, the objective number, be the size of neighborhood reference vectors. For one solution x_i, its neighborhood reference vectors is defined as $\mathrm{NRV}^i = u^{i,1}, u^{i,2}, \ldots, u^{i,M}$ where for each $u^j \notin \mathrm{NRV}^i$ and each $u^k \in \mathrm{NRV}^i$, $\langle x_i, u^j \rangle < \langle x_i, u^k \rangle$. In other words, the neighborhood reference vectors of x_i are the reference vectors with the first M smallest acute angle to x_i among all the reference vectors.

Definition 4 (angle position). The definition of angle position is designed to determine the spatial position of one solution relative to the reference vectors around it. For a solution x_i, NRV^i is the set of its neighborhood reference vectors. If $M = 2$, the angle position of x_i is defined as $\mathrm{AP}_i = |\langle x_i, \mathrm{NRV}^i_1 \rangle - \langle x_i, \mathrm{NRV}^i_2 \rangle|$. Generally speaking, the

angle position of a solution with biobjectives is the absolute value of the difference between the solution and its two neighborhood reference vectors. If $\mathrm{AP}_i = 0$, it means that the vector of the solution is in the middle of its two neighborhood reference vectors and the space between the two neighborhood reference vectors can be evenly divided by the vector of x_i. If $M > 2$, $\mathrm{AP}_i = \mathrm{Var}(\mathrm{NRV}_1^i, \mathrm{NRV}_2^i, \ldots, \mathrm{NRV}_M^i)$, the variance of the angles between X_i and its neighborhood reference vectors. Similarly, the vector of the solution with a smaller value of angle position is able to divide the space among the neighborhood reference vectors of the solution more evenly.

The definition of neighborhood reference vector is designed for a solution while the definition of neighborhood subspace is for a subspace. To visually describe the two definitions and the difference between them, a simple example is given. As can be seen in Figure 3, V_1, V_2, V_3, V_4, and V_5 are 5 reference vectors; X and Y are 2 solutions. A, B, C, and D are the angles between solutions and reference vectors. It is apparent that A is smaller than B and D is smaller than C; as a result, X is associated with V_2 and Y is associated with V_4. The neighborhood spaces of the subspace of V_2 are the corresponding subspaces of V_1 and V_3. The neighborhood reference vectors of X are V_2 and V_3. The angle position value of X is $|A - B|$ and that of Y is $|C - D|$. It is obvious that $|A - B| < |C - D|$ and the space between V_2 and V_3 can be more evenly divided by X when it is compared with the space between x and V_4 divided by Y.

4. The Proposed AP-RVEA

In this section, the details of the proposed AP-RVEA will be given. The overall framework of AP-RVEA is presented at first, followed by its main components.

4.1. The Main Framework of AP-RVEA. The pseudocode of the proposed AP-RVEA is shown in Algorithm 1. As shown in Algorithm 1, 4 parameters are first initialized in lines 1–4: (1) a set of uniformly distributed reference vectors V; (2) a population with N individuals; (3) the number of function evaluations: FEs; and (4) an archive in which elite individuals will be stored and will be updated in each generation. Then, the main loop of AP-RVEA is given in lines 5–12. In lines 6–7, the offspring population is generated and the FEs will be updated. Q is combination of the parent population and offspring population. In lines 9–10, when half of the optimization process pasts, the reference vectors will be adaptively adjusted via function Reference Vector Adjustment. The adjusted reference vector and the archive of elite individuals will be used to select promising individuals from the combined population Q. In line 12, the new archive of elite solutions will be generated according to the current archive and the new parent population P. The environmental selection strategy and the reference vectors adjustment strategy are the two main contributions of this paper and will be detailed in the following section.

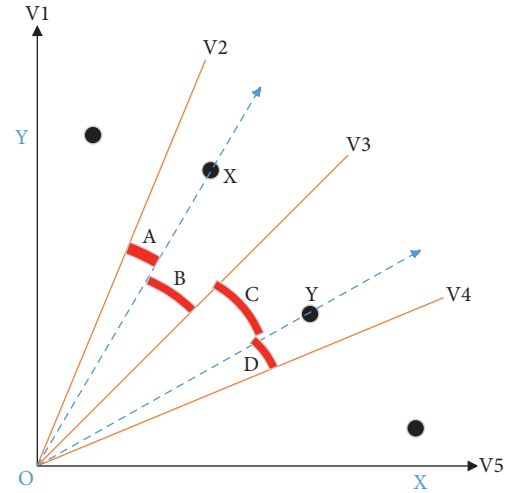


Figure 3: An example showing the definitions of subspace.

4.2. Reference Vector Adjustment. The pseudocode of function Reference Vector Adjustment is described in Algorithm 2. In this function, the inactive reference vectors (no solution exists in its corresponding subspace) will be deleted and new promising reference vectors will be generated from archive, the combination of elite solutions. The inputs of this function are current reference vectors V, the set of elite individuals archive, and the combine population Q. The output of this function refers to the updated reference vectors V.

At first, individuals in archive are normalized according the following equations:

$$\mathrm{archive}(i) = \frac{\mathrm{archive}(i) - Z^{\min}}{Z^{\max} - Z^{\min}}, \tag{4}$$

where $Z^{\min} = \{Z^{\min_1}, Z^{\min_2}, \ldots, Z^{\min_M}\}$ and $Z^{\max} = \{Z^{\max_1}, Z^{\max_2}, \ldots, Z^{\max_M}\}$ and $Z^{\min_i}, i \in \{1, 2, \ldots, M\}$, and $Z^{\max_i}$, $i \in \{1, 2, \ldots, M\}$, are the minimum value and maximal value of the ith objective among archive, respectively. The normalized $\mathrm{archive}(i)$ ranges from 0 to 1. After that, new reference vector is constructed as

$$\mathrm{new}\,A(i) = \frac{\mathrm{archive}(i)}{\sum_{j=1}^{M} \mathrm{archive}(i)_j}, \tag{5}$$

and through this transformation, $\sum_{i=1}^{M} \mathrm{new}\,A(i)_i = 1$.

Then, each individual in Q will be associated with an unit reference vector in V, i.e., each will lie in a subspace. In lines 6-7, the neighborhood reference vectors of each individual in new A are obtained and stored in NRV. In lines 8–10, the inactive reference vectors will be deleted from the current reference vectors. The addition of new reference vectors are described in lines 11–17. Since the new reference vectors are chosen from new A and the maximal number of reference vectors is N, once $|\mathrm{new}\,A|$ is 0 or $|V|$ is larger than N, the process will stop. In line 12, the AP (angle position) value of each individual in new A is initialized. In lines 13-14, the AP values of all the individuals will be calculated and stored. Among those individuals, the individual with the smallest AP value will be chosen as a new reference vector. Next, individuals

Input: population Size N; objective size M; the maximal function evaluations (MFEs);
Output: final population P;
(1) Generate a set of unit reference vectors $V = \{v_1, v_2, \ldots, v_N\}$;
(2) Generate a population $P \leftarrow$ Initialtization (N);
(3) Initialize function evaluations FEs$\leftarrow N$;
(4) archive$\leftarrow \varnothing$;
(5) **while** FEs < MFEs **do**
(6) $P' \leftarrow$ Generate Offspring;
(7) FEs$\leftarrow$FEs + N;
(8) $Q \leftarrow P \cap P'$;
(9) **if** (FEs/MFEs) > 50% **then**
(10) $V \leftarrow$ Reference Vector Adjustment $(V, Q,$ archive$)$;
(11) $P \leftarrow$ Environmental Selection (V, Q, N);
(12) archive$\leftarrow$ Archive Updat $(P,$ archive$)$;

Algorithm 1: The main procedure of AP-RVEA.

Input: V; archive; the combined population Q;
Output: updated V;
(1) **for** $i = 1 \longrightarrow$ |archive| **do**
(2) archive $(i) \leftarrow$ (archive $(i) - Z^{\min} / (Z^{\max} - Z^{\min})$);
(3) **for** $i = 1 \longrightarrow$ |archive| **do**
(4) new $A(i) \leftarrow$ (archive $(i) / \sum_{j=1}^{M}$ archive $(i)_j$);
(5) Associate each solution in Q with an unit reference vector in V according to Definition 2;
(6) **for** $i = 1 \longrightarrow$ |new A| **do**
(7) Find the neighborhood reference vectors of new A_i among V and store as NRV_i;
(8) **For** $i = 1 \longrightarrow |V|$ **do**
(9) **if** no solution is associated to V_i **then**
(10) $V \leftarrow V / V_i$;
(11) **while** |new A| > 0 and $|V| < N$ **do**
(12) AP$\leftarrow 0_{1,|\text{new } A|}$;
(13) **for** $i = 1 \longrightarrow$ |new A| **do**
(14) $\mathrm{AP}_i \leftarrow$ Calculate angle position value for new A_i;
(15) $k \leftarrow \arg\min \mathrm{AP}_i, i \in \{1, 2, \ldots, |\text{new } A|\}$;
(16) $V \leftarrow V \cup$ new A_k
(17) Remove all the solutions that own the same neighborhood reference vectors with new A_k in new A;

Algorithm 2: Reference vector adjustment.

that have the same neighborhood reference vectors will be removed from new A. That is to say, through each iteration, a new reference vector will be chosen and some individuals will be removed from new A, and once V has N reference vectors or new A is an empty set, this procedure will stop. It should be noted that removing some individuals from new A is a penalization mechanism. To visually show the detailed mechanism of this function, a simple example is plotted.

As can be seen in Figure 4(a), V_1, V_2, V_3, V_4, and V_5 denote the predefined unit reference vectors, and A, B, C, D, E, and F represent 6 solutions. The arrows in Figure 4(a) point from the solutions to their associated reference vectors, e.g., A is associated with V_1 and C is associated with V_2. At first, the AP values of the six solutions are calculated and the solution with the least AP value will be selected as the first new reference vector. In Figure 4(b), the solution A is chosen as the first new reference vector and there is no solution with the same neighborhood reference vectors as A that need to be removed (line 17 in Algorithm 2). Then, in Figure 4(c), the solution B is selected as the second new reference vector. In Figure 4(d), as the solution B and the solution C own the same neighborhood reference vectors (V_2 and V_3), solution C is removed from the candidate solution set. In Figure 4(e), the solutions E and D are selected or removed in the same way. Figure 4(f) shows the final result of the reference vectors after being modified by the reference vector adjustment strategy. It is obvious that the active subspace is further divided by the new reference vectors and may contribute more for the diversity performance of the population in the following environmental selection method. Besides, for a solution, its position in the objective space instead of corresponding reference vector matters whether it can be constructed as a new reference vector.

4.3. Environmental Selection. The environmental selection strategy of RVEA has two obvious weaknesses. (1) In $d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \|f'_{t,i}\|$, $|f'_{t,i}|$ works to distinguish the

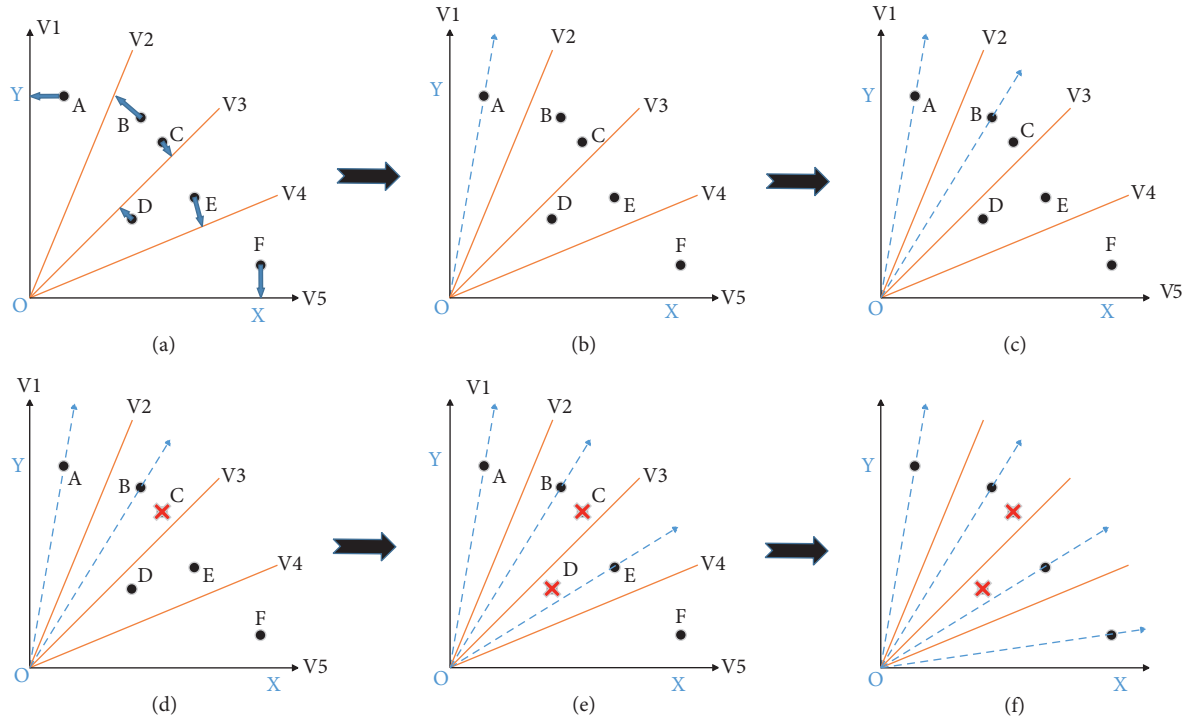


Figure 4: An example of the reference vector adjustment strategy.

convergence performance of a solution. However, when the number of objectives increases, the length of the objective vector is poor at posing enough selection pressure to the candidate population and selecting solutions with promising convergence performance. (2) After the APD method of each solution is calculated, RVEA selects one solution from each subspace to maintain sound diversity for the new population, while it is hard to guarantee that each reference vector can be associated with one or more solutions, especially with the increase of the number of objectives. To cope with the above two issues, the proposed AP-RVEA offers two methods.

We choose to construct a new parameter to replace $f_{t,i}'$ to evaluate the convergence performance in $d_{t,i,j}$. As can be seen in Algorithm 3, all the solutions in Q are first normalized in lines 1–4. Then, each solution is associated with a subspace in lines 5–7. P_i means the solutions in the ith subspace. Inspired by fractional dominance relation [35], if x_i lies in P_j, the new parameter can be constructed as follows:

$$fv(i) = \sum_{k=1}^{|P_j|} \sum_{q=1}^{M} |\{q\} | x_i^q < P_{j,k}^q|,$$
$$fv'(i) = \frac{fv(i)}{\sum_{g=1}^{|P_j|} fv\left(P_{j,g}\right)}, \tag{6}$$

where $P_{j,k}$ means the kth solution in P_j and $P_{j,k}^q$ means the qth objective of $P_{j,k}$. $fv(i)$ denotes the sum of the number of objectives on which x_i surpasses each solution that lies in the same subspace with x_i. $fv'(i)$ is a normalized value of $fv(i)$ and ranges from 0 to 1. For two solutions x_i and x_j in the same subspace, if $fv'(i) > fv'(j)$, x_i surpasses x_j in terms of convergence performance. It is worth noting that the calculation of fv' in different subspaces is independent of each other and thus offers a good way to guarantee that solutions with good convergence performance in each subspace can be found and selected. The improved APD can be calculated as $d_{t,i,j}' = (1 + P(\theta_{t,i,j})) \cdot (1/fv'(t,i))$, where $|f_{t,j}'|$, the length of the normalized solution, is replaced by the aforementioned new parameter. With the improved APD method, the shortage of the original APD method that it is poor at posing enough selection pressure can be filled.

To make the definition of $fv(i)$ more clear, an example is shown to make the readers easily understand $fv(i)$. x_1, x_2, x_3, x_4, and x_5 are 5 objective vectors. x_1, x_2, and x_3 lie in the first subspace, and x_4 and x_5 are in the second subspace. x_1 surpasses x_2, x_3, x_4, and x_5 on 3, 4, 2, and 6 objectives, respectively. Then, the value $fv(x_1)$ is 7, the sum of the objective on which x_1 surpasses x_2 and x_3. The number of the objectives on which x_1 surpasses x_4 and x_5 is not used in the calculation of $fv(x_1)$ because x_4 and x_5 lie in different subspaces with x_1. The same way can be used to calculate $fv(x_2)$, $fv(x_3)$, $fv(x_4)$, and $fv(x_5)$.

After the improved APD methods of population Q are calculated in line 8 in Algorithm 3, the method tailored for addressing the second issue mentioned above is described in lines 9–13. The loop will continue till the population size increases to N: in line 12, one solution will with the least improved APD method will be selected into P in each subspace and these solutions will be removed from Q in line 13; then, if the size of P still less than N, the remaining solutions in Q will be selected into P in the same way as the selected solutions until $|P| = N$.

Input: V; population size N; the combined population Q;
Output: new population P;
(1) $P \leftarrow \varnothing$
(2) Calculate the ideal point $Z^{\min}$ for Q
(3) **for** $i = 1 \longrightarrow |Q|$ **do**
(4) $\quad f_i \leftarrow f_i - Z^{\min}$
(5) **for** $i = 1 \longrightarrow |Q|$ **do**
(6) $\quad k \leftarrow \text{argmin}_{j \in \{1,2,\ldots,|V|\}} \langle Q_i, V_j \rangle$
(7) $\quad P_k \leftarrow P_k \cup Q_i$
(8) Calculate the improved APD method for each solution in $P_k, k = \{1, 2, \ldots, |V|\}$
(9) **while** $|P| < N$ **do**
(10) $\quad$ **for** $i = 1 \longrightarrow |V|$ **do**
(11) $\quad\quad$ **if** $|P_i| > 0$ **then**
(12) $\quad\quad\quad$ Selected the solution with the minimum improved APD method into P;
(13) $\quad\quad\quad$ Remove the solution from P_i;

Algorithm 3: Environmental selection.

4.4. Archive Update. The pseudocode of function ArchiveUpdate is shown in Algorithm 4. The inputs of this function are the current archive, population size N, and the new parent population P. In line 1, the dominated solutions in P will be filtered out and the remaining solutions will be stored in P_{ND}. If $|\text{archive}| + |P_{\text{ND}}|$ is larger than N, some solutions in the current archive will be removed until the size of combination of archive and P_{ND} is equal to the population size, the predefined size of archive. If the size of the combination of P_{ND} and the current archive is less than N, all solutions in P_{ND} will be absorbed into archive. The updated criterion is that nondominated solutions in current generation are superior to the nondominated solutions in the last generation, so the excess nondominated solutions in last generation will be randomly removed. In each generation, archive will be updated.

5. Experimental Studies and Discussion

To test the performance of the proposed AP-RVEA, we compare it with 5 state-of-the-art algorithms: A-NSGA-III [32], MOEA/D [5], RVEA* [16], MaOEA-IT [36], and MOEA/D-PaS [37]. The proposed AP-RVEA and the comparative algorithms are all encoded in MATLAB and embedded in PlatEMO [38], which is free to public. All the experiments are run on MATLAB R2018a.

5.1. Experimental Settings

5.1.1. Benchmark Problems. The performance of the six algorithms is compared in the context of MaF1 to MaF9 taken from MaF test suite [18], with 3, 5, 8, and 10 objectives. These 9 benchmarks contain various properties, e.g., disconnected, multimodal, irregular PF, deceptiveness, etc. In this section, a benchmark with a specific number of objective is referred to as a test instance.

5.1.2. Performance Indicator. The hypervolume [39] (HV) and inverted generational distance [40] are chosen to evaluate the final population output by the six algorithms.

(1) The HV is the volume of the space consisting of a reference point and a set of solutions and is widely used to reflect the performance of a population in terms of both convergence and diversity. A population with a larger HV value tends to own better overall performance. In this paper, the reference point used to calculate the HV value of a population on each test instance is embedded in PlatEMO. Besides, in this paper, all the HV values are normalized to $[0, 1]$.

(2) IGD evaluates the overall performance of a population by calculating the distance between the population and the true Pareto front. During the process, the true Pareto front is represented by a set of solutions lying in it and the number in PlatEMO is set to 10,000. The IGD can be calculated as follows:

$$\text{IGD}\left(P^*, P\right) = \frac{\sum_{v \in P*} D_{\min}(v, P)}{\left|P^*\right|}, \tag{7}$$

where P^* is a set of reference points and P denotes the objective vectors of a population. $D_{\min}(v, P)$ represents the minimum Euclidean distance from the point v to all the points in P.

5.1.3. Termination Condition. The maximal number of function evaluations (MFEs) is adopted as the termination condition for the six algorithms. For MaF1–MaF9 with 3, 5, 8, and 10 objectives, the MFEs is set to 100,000.

5.1.4. Population Size. For MOEA/D [5] and other decomposition-based algorithms, the population size is largely determined by the total number of reference points in an M-objective problem. For problems with $M > 8$, a two-layer vector generation strategy can be employed to generate reference (or weight) vectors not only on the outer boundaries but also on the inside layers of the Pareto fronts [7]. Therefore, the population sizes of the six algorithms on

Input: archive; population size N; selected population P;
Output: updated archive;
(1) Filter out the dominated solutions in P and store the remaining solutions into P_{ND}
(2) **if** |archive| + $|P_{ND}| > N$ **then**
(3) $k \leftarrow$|archive| + $|P_{ND}| - N$;
(4) Randomly delete k solutions from archive; archive$\leftarrow$archive $\cup P_{ND}$
(5) **else**
(6) archive$\leftarrow$archive $\cup P_{ND}$

Algorithm 4: Archive update.

MaF1–MaF9 are set according to the number of objectives, that is, 100, 212, 156, and 275 for 3, 5, 8, and 10 objectives, respectively.

5.2. Environmental Results and Analyses. All the test instances are run 30 times, and the Wilcoxon rank-sum test with $\alpha = 0.5$ is applied to test the significant differences between the HV indicators of populations output by the six algorithms. The mean and standard deviation of HV values and IGD values are shown in Tables 1 and 2, respectively. The symbols −, +, and ≈ in the two tables mean that the corresponding HV values or IGD values are worse than, better than, and equal to that of AP-RVEA. In the bottom of Tables 1 and 2, the number of test instances on which each comparative algorithm behaves worse than, better than, and equal to AP-RVEA is counted.

As can be seen in Table 1, the proposed AP-RVEA surpasses all the comparative algorithms on 17 out of 36 test instances, while the numbers of that of A-NSGA-III, MOEA/D, RVEA*, MaOEA-IT, and MOEA/D-PaS are 5, 3, 7, 0, and 4. Besides, AP-RVEA outperforms the five comparative algorithms on 23, 23, 20, 36, and 23 test instances in terms of HV metric, respectively. In summary, the proposed AP-RVEA shows the best overall performance among the six algorithms. With respect to the results measured by IGD metric shown in Table 2, although the specific statistic are different from that of Table 1, it is also obvious that the proposed AP-RVEA shows promising overall performance. In more detail, AP-RVEA performs better than A-NSGA-III, MOEA/D, RVEA*, MaOEA-IT, and MOEA/D-Pas on 23, 22, 21, 32, and 27 test instances in all, respectively.

The reason why there exist some differences between Tables 1 and 2 may lie in the way the two metrics are calculated. HV metric evaluates the overall performance of a population according to the volume of the space decided by a predefined reference point and the nondominated solutions in the population. Consequently, the HV value is affected by the convergence of each solution in the population on each objective. Generally, it is a concrete value rather than a relative value, while IGD metric assesses the overall performance of a population by calculating the distance between the population and the true Pareto front. During the process, a set of uniformly distributed reference points is used to work as an approximation of the true PF during the calculation of IGD results. When the number of objectives is small, the reference points can be seen as a good approximation of the true PF with respect to uniformity and density; however, when the number of objectives increases, the objective space grows rapidly and the number of the reference points (in PlatEMO, the number is 10,000) is not enough to represent the whole PF of the problems. As a result, some errors may appear during the calculation process of IGD values. For example, if a few solutions in a population happen to lie very close to some reference points, then their contribution to the population IGD value will also be greater than some solutions with good convergence but far away from the reference point. After all, the objective space of MaOPs is very large. It is difficult or even impossible to guarantee that 10,000 reference points can be densely and evenly distributed on the entire PF to fairly evaluate the overall performance of a population by calculating the distances between it and the reference points. Furthermore, how many points can be said to be densely distributed on the real PF of a many-objective problem is still a problem.

Considering the above issues, the IGD and HV metrics may show contradictions in some test instances. However, the overall results shown by the two metrics are consistent, that is, the proposed AP-RVEA obtains the most competitive performance.

Such superior performance of AP-RVEA can be attributed to the following facts. Firstly, the reference vectors are adaptively adjusted according to the obtained solutions with promising performance, i.e., the approximated PF of the MaOP, and thus sound diversity of the obtained population can be achieved. Besides, after the reference vectors are adjusted, the convergence performance of the solutions in each subspace can be more accurately evaluated by an improved APD method compared with the original APD method. Generally, the reference vector adjustment strategy and the improved APD method work together to get the whole performance of AP-RVEA.

In order to visually show the comparison among the six algorithms, their output populations with the largest HV values among the 30 runs on 10-objective MaF1 and 8-objective MaF5 are plotted in the parallel coordinates. Figure 5 gives the comparison results of 10-objective MaF1, and Figure 6 shows the results of 8-objective MaF5.

MaF1 is a linear problem where no single optimal solution lies in any subset of objectives [18]. For 10-objective MaF1, the value of each objective ranges in $[0, 1]$. As can be seen in Figure 5(c), we can find that only a few solutions are output by RVEA*. For A-NSGA-III and MOEA/D-Pas, they fail to converge to the true PF on some objectives. In the

Table 1: HV results of the 6 algorithms on benchmarks MaF–MaF9.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA*[16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF1	3	2.0353e−1	1.9823e−1	2.1216e−1	8.0588e−2	1.8407e−1	2.1699e−1
		(1.24e−3) −	(1.76e−6) +	(1.81e−3) −	(4.47e−2) −	(1.07e−3) −	(5.89e−4)
	5	8.3089e−3	1.1127e−2	4.0294e−3	2.6586e−3	5.4484e−3	6.1155e−3
		(1.82e−4) +	(3.73e−4) +	(4.33e−4) −	(2.46e−3) −	(1.01e−4) −	(5.10e−4)
	8	1.1271e−5	3.1161e−6	7.0858e−7	7.5799e−7	2.6431e−5	2.6546e−6
		(3.71e−6) −	(9.84e−7) −	(4.00e−7) −	(1.56e−6) −	(1.73e−6) −	(1.19e−6)
	10	7.9566e−9	1.3977e−8	5.7247e−9	2.6723e−9	2.3472e−8	6.4159e−8
		(4.94e−9) −	(1.82e−9) −	(3.03e−9) −	(5.03e−9) −	(3.55e−8) −	(1.93e−8)
MaF2	3	2.2882e−1	2.3886e−1	2.4321e−1	1.3382e−1	2.3105e−1	2.2968e−1
		(5.62e−4) −	(7.14e−4) +	(9.30e−4) +	(1.83e−2) −	(1.28e−3) +	(1.52e−3)
	5	1.8268e−1	1.8704e−1	1.7520e−1	1.3101e−1	9.2288e−2	1.9060e−1
		(1.23e−3) −	(4.46e−4) −	(1.92e−3) −	(3.22e−2) −	(6.42e−3) −	(1.74e−3)
	8	2.0533e−1	2.0719e−1	1.7835e−1	1.4836e−1	7.4638e−2	1.7799e−1
		(3.14e−3) +	(6.02e−4) +	(5.28e−3) ≈	(3.23e−2) −	(1.84e−2) −	(4.91e−3)
	10	1.0651e−1	1.7083e−1	1.0762e−1	9.2650e−2	4.3372e−2	1.7975e−1
		(9.50e−3) −	(7.94e−4) −	(9.34e−3) −	(1.41e−2) −	(3.33e−4) −	(7.45e−3)
MaF3	3	9.5716e−1	9.5491e−1	9.6108e−1	8.4815e−2	7.1020e−1	9.5871e−1
		(6.89e−4) −	(1.13e−3) −	(8.13e−4) +	(2.60e−1) −	(3.76e−1) −	(1.76e−3)
	5	9.4447e−1	9.8725e−1	8.2530e−1	0.0000e+0	2.1056e−1	9.9929e−1
		(2.12e−1) −	(1.44e−3) −	(3.35e−1) −	(0.00e+0) −	(3.80e−1) −	(3.38e−4)
	8	7.8110e−1	9.6767e−1	9.9682e−1	0.0000e+0	9.5248e−3	9.9939e−1
		(3.94e−1) −	(2.78e−3) −	(3.04e−4) −	(0.00e+0) −	(2.85e−2) −	(2.13e−2)
	10	9.99454e−1	9.4948e−1	9.9851e−1	0.0000e+0	2.8415e−3	9.1280e−1
		(1.28e−2) +	(2.78e−3) +	(4.61e−4) +	(0.00e+0) −	(1.56e−2) −	(2.72e−1)
MaF4	3	5.2149e−1	5.0055e−1	5.2054e−1	4.6992e−2	4.2732e−1	4.5632e−1
		(6.16e−3) +	(3.40e−3) ≈	(2.85e−3) +	(1.37e−1) −	(1.80e−1) ≈	(8.95e−2)
	5	5.8544e−2	7.4840e−3	3.4932e−2	0.0000e+0	4.2368e−2	6.0373e−2
		(5.49e−3) ≈	(2.94e−3) −	(1.10e−2) −	(0.00e+0) −	(2.20e−2) −	(1.06e−2)
	8	3.8552e−5	2.5638e−6	3.9599e−5	0.0000e+0	2.8057e−3	1.7934e−5
		(1.55e−5) +	(3.90e−7) −	(3.54e−5) +	(0.00e+0) −	(1.47e−4) +	(1.69e−5)
	10	3.0500e−8	1.4597e−8	2.8142e−7	1.1961e−7	2.0046e−4	1.8383e−7
		(2.50e−8) −	(1.86e−9) −	(2.42e−7) +	(6.55e−7) −	(4.62e−5) +	(1.28e−7)
MaF5	3	4.9687e−1	4.7843e−1	4.5833e−1	1.0907e−1	4.5198e−1	5.5958e−1
		(9.09e−2) −	(1.35e−1) −	(1.72e−1) −	(3.90e−2) −	(1.56e−1) −	(3.09e−5)
	5	7.3778e−1	4.4689e−1	8.1174e−1	1.2033e−1	7.4462e−1	8.0819e−1
		(2.33e−2) −	(1.05e−1) −	(4.40e−4) +	(9.19e−2) −	(3.40e−2) −	(2.53e−2)
	8	8.2333e−1	3.5794e−1	8.9065e−1	2.5990e−2	8.9836e−1	8.9915e−1
		(1.43e−2) −	(8.63e−2) −	(4.49e−2) −	(4.18e−2) −	(6.10e−3) ≈	(6.73e−3)
	10	8.7108e−1	2.5125e−1	8.2530e−1	6.7036e−2	8.7214e−1	8.0365e−1
		(2.20e−2) +	(7.94e−2) −	(9.06e−2) ≈	(4.50e−2) −	(1.21e−1) +	(2.30e−2)
MaF6	3	1.9257e−1	1.8185e−1	1.8865e−1	4.6163e−2	1.8937e−1	1.7995e−1
		(1.19e−3) +	(5.24e−6) +	(1.50e−3) +	(1.07e−2) −	(8.73e−4) +	(1.09e−3)
	5	1.2878e−1	1.1532e−1	1.0692e−1	1.0711e−1	1.0778e−1	1.3141e−1
		(4.32e−4) −	(3.49e−2) −	(1.46e−2) −	(4.43e−2) −	(8.34e−4) −	(9.79e−3)
	8	7.5412e−2	9.5638e−2	9.4549e−2	8.3754e−2	6.3906e−2	9.5782e−2
		(4.26e−2) −	(2.45e−2) −	(3.40e−3) −	(3.75e−2) −	(3.70e−2) −	(1.33e−3)
	10	8.4470e−2	7.1992e−2	9.4688e−2	9.0951e−3	8.1006e−2	8.5933e−2
		(5.98e−3) −	(4.28e−2) ≈	(1.14e−3) ≈	(2.77e−2) −	(2.76e−2) ≈	(1.23e−2)
MaF7	3	2.5256e−1	2.5331e−1	2.6855e−1	1.2596e−1	2.1428e−1	2.5443e−1
		(5.32e−3) −	(1.38e−2) ≈	(1.22e−2) +	(3.78e−2) −	(9.79e−2) ≈	(5.81e−3)
	5	2.1329e−1	1.4613e−1	2.2129e−1	4.9265e−3	2.1578e−1	2.4446e−1
		(8.04e−3) −	(1.31e−2) −	(3.85e−3) −	(9.10e−3) −	(6.01e−2) −	(2.11e−3)
	8	9.2868e−2	3.9645e−3	1.8451e−1	0.0000e+0	1.8405e−1	1.2609e−1
		(1.71e−2) −	(1.39e−2) −	(2.99e−3) +	(0.00e+0) −	(7.74e−3) +	(1.95e−2)
	10	1.3202e−1	5.3985e−5	1.0158e−1	0.0000e+0	1.5212e−3	6.6357e−2
		(1.55e−2) +	(1.47e−4) −	(4.76e−2) +	(0.00e+0) −	(8.33e−3) −	(1.80e−2)

Table 1: Continued.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA* [16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF8	3	2.5863e−1 (2.33e−3) +	2.5984e−1 (6.33e−3) +	2.6481e−1 (2.65e−3) +	2.4228e−2 (3.38e−2) −	0.0000e+0 (0.00e+0) −	2.4098e−1 (3.60e−3)
	5	1.0867e−1 (1.69e−3) −	1.1426e−1 (2.34e−3) −	8.8484e−2 (5.55e−3) −	6.8305e−3 (1.08e−2) −	0.0000e+0 (0.00e+0) −	1.1673e−2 (5.91e−3)
	8	2.4678e−2 (6.97e−4) +	2.2108e−2 (4.52e−4) +	1.1092e−2 (3.72e−3) −	8.5991e−4 (1.66e−3) −	0.0000e+0 (0.00e+0) −	1.5505e−2 (1.89e−3)
	10	3.8703e−3 (7.50e−4) −	6.2901e−3 (2.08e−4) −	2.6006e−3 (8.94e−4) −	4.9137e−5 (1.52e−4) −	0.0000e+0 (0.00e+0) −	7.9081e−3 (2.71e−4)
MaF9	3	8.3695e−1 (1.26e−3) −	8.3183e−1 (1.52e−2) −	8.4344e−1 (9.81e−4) −	9.9304e−3 (5.44e−2) −	7.7608e−1 (1.48e−1) −	8.5828e−1 (4.12e−2)
	5	2.8542e−1 (3.37e−3) +	3.1093e−1 (2.91e−3) +	2.3391e−1 (1.09e−2) −	0.0000e+0 (0.00e+0) −	2.3636e−1 (5.28e−2) ≈	2.5743e−1 (1.39e−2)
	8	3.3482e−2 (5.60e−3) +	4.1322e−2 (5.03e−4) +	2.2144e−2 (2.52e−3) +	0.0000e+0 (0.00e+0) −	4.4560e−2 (3.99e−4) +	1.9959e−2 (2.68e−3)
	10	6.2283e−3 (1.26e−3) −	1.1816e−2 (2.24e−3) −	4.6773e−3 (1.18e−3) −	0.0000e+0 (0.00e+0) −	1.2047e−2 (2.47e−3) ≈	1.2064e−2 (4.20e−4)
−/+/ ≈		23/12/1	23/10/3	20/13/3	36/0/0	23/7/6	

Table 2: IGD results of the 6 algorithms on benchmarks MaF–MaF9.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA* [16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF1	3	6.2920e−2 (1.58e−3) +	7.0475e−2 (7.51e−7) +	5.3794e−2 (2.18e−3) +	2.8372e−1 (1.02e−1) −	9.0179e−2 (1.35e−3) −	8.0285e−2 (1.44e−3)
	5	2.5753e−1 (1.67e−2) −	1.2684e−1 (5.62e−3) +	1.8403e−1 (1.53e−2) −	3.1034e−1 (9.67e−2) −	2.2529e−1 (1.48e−3) −	1.6647e−1 (3.70e−3)
	8	3.5534e−1 (3.65e−2) −	4.6348e−1 (2.69e−2) −	5.9242e−1 (6.12e−2) −	5.5330e−1 (1.21e−1) −	5.0319e−1 (6.46e−2) −	2.7481e−1 (8.18e−3)
	10	6.1235e−1 (7.22e−2) −	5.4023e−1 (1.77e−2) −	6.3868e−1 (7.02e−2) −	6.0878e−1 (1.39e−1) −	3.3410e−1 (1.03e−2) +	4.3303e−1 (2.73e−2)
MaF2	3	3.6496e−2 (3.86e−4) +	3.8937e−2 (5.45e−4) +	2.8213e−2 (6.58e−4) +	1.4382e−1 (3.30e−2) −	5.4623e−2 (1.56e−3) −	4.2871e−2 (1.24e−3)
	5	1.1526e−1 (8.77e−4) −	1.1107e−1 (2.19e−4) −	8.7126e−2 (1.47e−3) −	1.3455e−1 (5.03e−2) ≈	2.2794e−1 (1.03e−2) −	6.1168e−2 (1.19e−3)
	8	1.9101e−1 (4.89e−3) +	2.1724e−1 (3.90e−4) +	1.7075e−1 (4.24e−3) +	3.1336e−1 (5.46e−2) +	2.1058e−1 (2.56e−3) +	7.8364e−1 (3.26e−2)
	10	3.9322e−1 (7.91e−2) −	2.6599e−1 (1.11e−3) −	2.1699e−1 (5.71e−2) −	4.4354e−1 (5.77e−2) −	8.6610e−1 (1.65e−3) −	2.1087e−1 (5.37e−3)
MaF3	3	4.7122e−2 (5.65e−4) −	5.1895e−2 (1.84e−3) −	4.1828e−2 (1.24e−3) ≈	3.8136e+2 (1.48e+3) −	1.5467e+2 (8.31e+2) −	4.2030e−2 (2.13e−3)
	5	8.7541e−1 (2.03e+0) −	1.0687e−1 (4.15e−3) −	6.9057e−1 (1.97e−3) −	1.5492e+4 (7.55e+4) −	2.0454e+3 (5.63e+3) −	1.3820e−1 (2.61e−1)
	8	9.3781e−1 (2.79e+0) −	1.6283e−1 (1.88e−3) −	1.0497e−1 (1.47e−2) −	2.0367e+7 (8.77e+7) −	1.2781e−1 (4.10e−2) −	1.0014e−1 (1.65e+4)
	10	1.0131e−1 (2.33e−2) +	1.3934e−1 (1.33e−3) +	9.6908e−2 (4.41e−3) +	3.0285e+8 (1.10e+9) −	1.5765e+8 (5.03e+8) −	2.2075e−1 (3.61e−1)
MaF4	3	4.0702e−1 (1.50e−2) ≈	6.7751e−1 (2.65e−2) −	3.3615e−1 (1.11e−2) +	5.0300e+1 (1.15e+2) −	9.9331e−1 (1.74e+0) ≈	5.5332e−1 (3.32e−1)
	5	3.4214e+0 (1.07e+0) −	1.0289e+1 (7.60e−1) −	2.8889e+0 (4.54e−1) −	4.3480e+2 (8.85e+2) −	1.4022e+1 (3.01e+1) −	2.4721e+0 (9.11e−2)
	8	3.0978e+1 (3.13e+0) ≈	1.1789e+2 (6.00e+0) −	3.4092e+1 (1.45e+0) −	2.0335e+3 (3.12e+3) −	5.1337e+1 (1.53e+1) −	2.9338e+1 (1.04e+0)
	10	2.5694e+2 (6.10e+1) −	5.3290e+2 (4.20e+1) −	1.0044e+2 (9.25e+0) +	8.8107e+3 (1.43e+4) −	1.2975e+2 (2.65e+1) +	1.9903e+2 (3.41e+1)
MaF5	3	6.4826e−1 (6.61e−1) −	8.5931e−1 (1.24e+0) −	1.1243e+0 (1.62e+0) −	4.6662e+0 (3.11e−1) −	9.8315e−1 (1.17e+0) −	2.5976e−1 (2.71e−5)
	5	1.9720e+0 (4.73e−3) +	8.1263e+0 (1.43e+0) −	1.7716e+0 (1.22e−1) +	1.0212e+1 (2.46e+0) −	2.5855e+0 (2.42e−1) ≈	2.4960e+0 (6.21e−2)
	8	3.0063e+1 (1.81e+0) −	8.3568e+1 (1.46e+0) −	2.2528e+1 (2.63e+0) −	8.9764e+1 (5.22e+1) −	2.6183e+1 (3.39e+0) −	2.1817e+1 (9.14e−1)
	10	1.3002e+2 (3.12e+1) −	3.0356e+2 (1.66e+0) −	1.1822e+2 (2.80e+1) −	2.3986e+2 (4.90e+1) −	1.4810e+2 (2.51e+1) −	1.0322e+2 (1.15e+1)

Table 2: Continued.

	M	A-NSGA-III [32]	MOEA/D [5]	RVEA* [16]	MaOEA-IT [36]	MOEA/D-PaS [37]	AP-RVEA
MaF6	3	1.4787e − 2	3.3929e − 2	2.2360e − 2	3.2842e − 1	2.3682e − 2	3.4808e − 2
		(1.49e − 3) +	(4.12e − 6) ≈	(2.67e − 3) +	(6.47e − 2) −	(1.53e − 3) +	(2.06e − 3)
	5	7.5945e − 2	5.4923e − 2	2.2218e − 1	7.6317e − 2	9.6494e − 2	9.1551e − 3
		(8.31e − 3) −	(9.94e − 2) −	(2.66e − 1) −	(1.65e − 1) −	(1.53e − 2) −	(3.10e − 3)
	8	1.4803e − 1	1.8737e − 1	3.7356e − 1	7.2548e − 2	1.1124e − 1	8.3171e − 1
		(1.71e − 1) +	(2.34e − 1) +	(2.84e − 1) +	(1.60e − 1) +	(2.02e − 2) +	(1.91e − 1)
	10	1.5499e − 1	2.6580e − 1	2.3930e − 1	4.4674e − 1	2.4357e + 1	1.7272e − 1
		(4.34e − 2) ≈	(2.54e − 1) ≈	(2.27e − 1) ≈	(6.27e − 2) −	(7.21e + 1) −	(1.26e − 1)
MaF7	3	1.0095e − 1	1.9739e − 1	1.2263e − 1	5.2536e − 1	1.1855e + 0	1.1628e − 1
		(4.88e − 2) +	(1.64e − 1) −	(1.04e − 1) −	(1.64e − 1) −	(2.31e + 0) −	(7.04e − 3)
	5	4.9241e − 1	5.6292e − 1	3.7725e − 1	3.5548e + 0	9.6303e − 1	3.7503e − 1
		(9.87e − 3) −	(1.78e − 1) −	(1.17e − 2) −	(2.52e + 0) −	(1.04e + 0) −	(1.42e − 2)
	8	1.2508e + 0	1.7909e + 0	1.6928e + 0	2.2385e + 1	1.8109e + 0	2.6202e + 0
		(7.13e − 2) +	(3.49e − 1) +	(1.78e − 1) +	(2.24e + 0) −	(1.24e − 1) +	(5.54e − 1)
	10	2.5671e + 0	2.7215e + 0	1.1068e + 0	3.6904e + 1	1.9033e + 1	1.7752e + 0
		(3.68e − 1) −	(4.95e − 1) −	(1.01e − 1) +	(2.06e + 0) −	(6.97e + 0) −	(1.58e − 1)
MaF8	3	1.1078e − 1	1.1047e − 1	9.2711e − 2	1.6084e + 2	4.0405e + 1	1.3668e − 1
		(4.88e − 3) +	(1.59e − 2) +	(5.08e − 3) +	(2.85e + 2) −	(2.44e + 1) −	(8.50e − 3)
	5	2.6739e − 1	2.0045e − 1	2.9639e − 1	6.5221e + 1	2.8168e + 1	1.6703e − 1
		(2.24e − 2) −	(3.08e − 2) −	(7.17e − 2) −	(9.04e + 1) −	(2.21e + 1) −	(1.08e − 2)
	8	4.6062e − 1	6.8716e − 1	8.8528e − 1	1.5075e + 2	6.8703e − 1	4.8632e + 1
		(5.26e − 2) +	(1.97e − 2) +	(1.77e − 1) +	(1.87e + 2) ≈	(7.28e − 2) +	(3.46e + 1)
	10	9.4959e − 1	9.6884e − 1	1.1676e + 0	2.1211e + 4	3.1499e + 1	6.3354e − 1
		(1.21e − 1) −	(2.09e − 2) −	(2.13e − 1) −	(1.34e + 4) −	(2.91e + 1) −	(5.75e − 2)
MaF9	3	6.2530e − 2	6.4715e − 2	5.4752e − 2	6.0017e + 1	9.9649e − 1	5.0301e − 2
		(1.31e − 3) −	(5.85e − 3) −	(6.20e − 4) −	(7.30e + 1) −	(4.90e + 0) −	(7.12e − 2)
	5	2.7053e − 1	9.9632e − 2	2.0823e − 1	8.5660e + 1	4.7588e − 1	1.6044e − 1
		(2.81e − 2) −	(3.22e − 3) +	(3.35e − 2) −	(1.37e + 2) −	(1.09e + 0) −	(8.21e − 3)
	8	3.1743e − 1	2.7830e − 1	6.1440e − 1	1.5605e + 2	6.3322e − 1	1.9853e − 1
		(1.55e − 1) −	(1.79e − 2) −	(9.20e − 2) −	(1.05e + 2) −	(8.37e − 2) −	(2.97e − 3)
	10	8.3765e − 1	6.5459e − 1	9.8642e − 1	3.9581e + 2	8.1119e − 1	3.8017e − 1
		(1.55e − 1) −	(5.98e − 1) −	(1.79e − 1) −	(4.97e + 2) −	(1.95e + 0) −	(2.27e − 2)
−/+/ ≈		23/10/3	22/12/2	21/13/2	32/2/2	27/7/2	

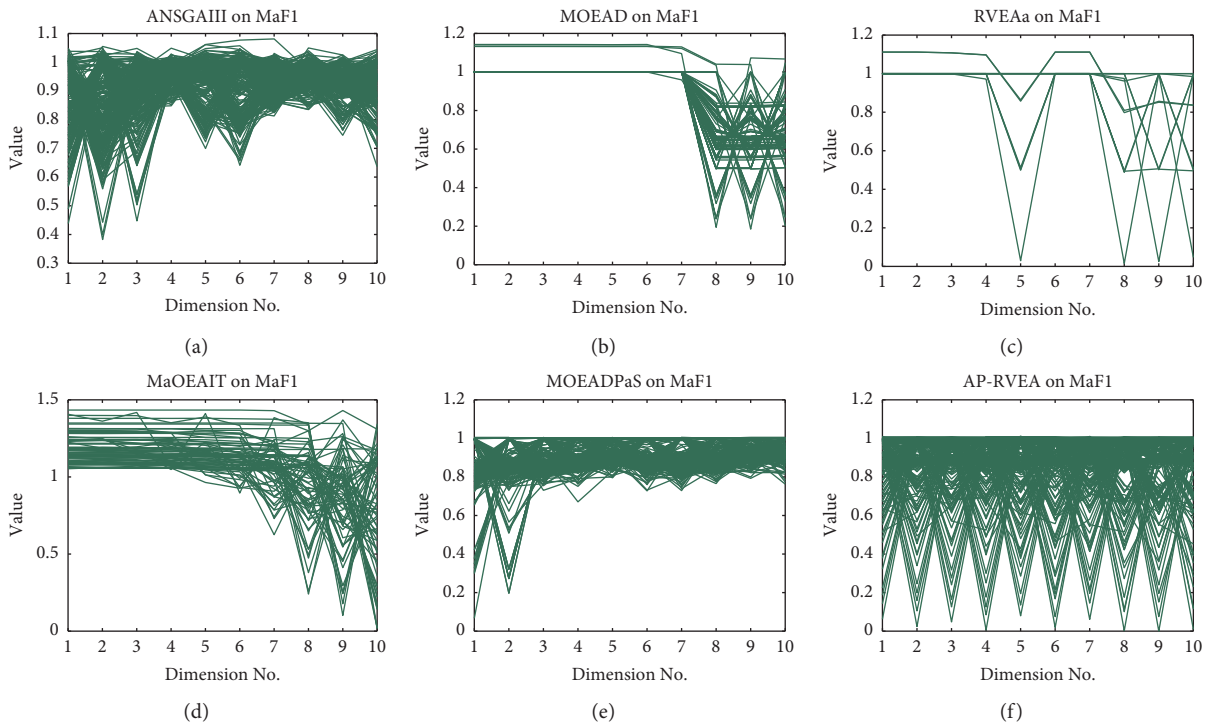


Figure 5: The best populations obtained by the six algorithms on 10-objective MaF1, shown by parallel coordinates.

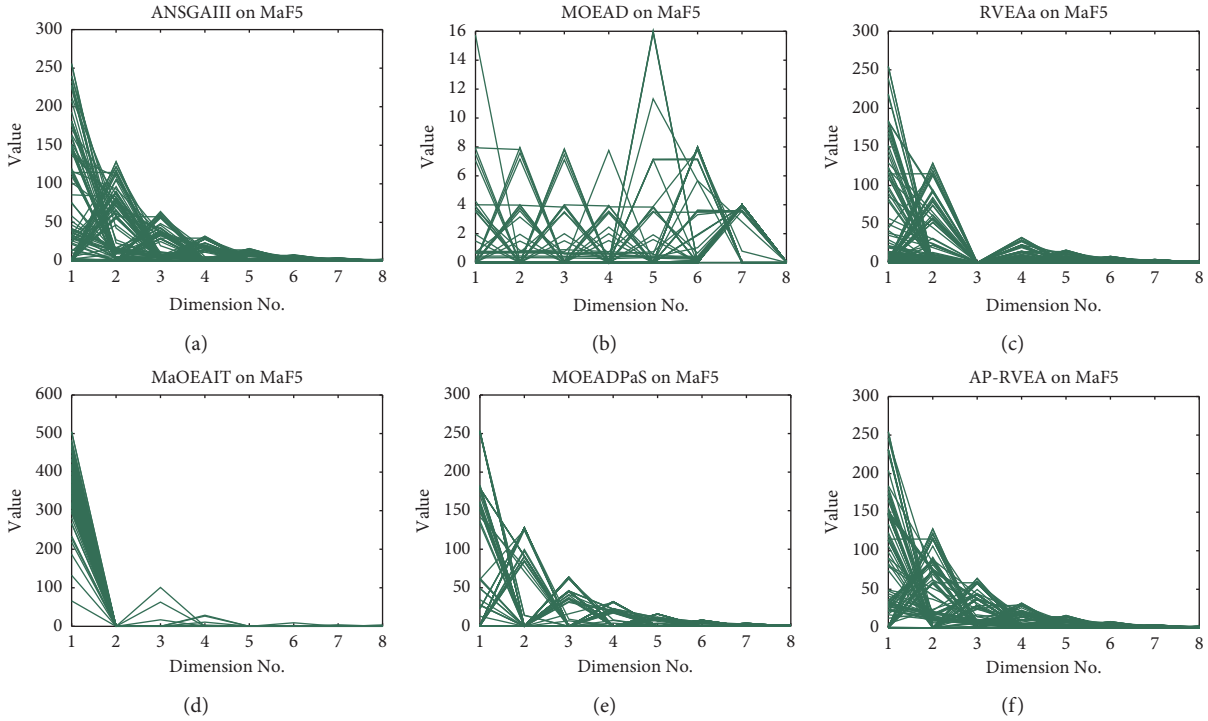


Figure 6: The best populations obtained by the six algorithms on 8-objective MaF5, shown by parallel coordinates.

populations output by MaOEA-IT and MOEA/D, it is obvious that some objectives of some solutions do not fall into $[0, 1]$. The proposed AP-RVEA gains good balance between convergence and diversity according to the figure of its population. In summary, AP-RVEA possesses the best overall performance on 10-objective MaF1.

The output populations obtained by the six algorithms on 8-objective MaF5 are plotted in Figure 6. MaF5 is a concave and biased problem and has a badly scaled PF. Each objective function of MaF5 is scaled to a substantially different range [18]. In Figures 6(a), 6(e), and 6(f), A-NSGA-III, MOEA/D-Pas, and AP-RVEA gain good approximations of the true PF of 8-objective MaF5, while RVEA* fails to converge on the third objective. For MaOEA-IT, we can see that the range of its output population is $[0, 500]$ instead of $[0, 250]$. Besides, the population obtained by MOEA/D seems to have a bad diversity as well as convergence according to the messy line in Figure 6. The results in Table 1 demonstrate that the proposed AP-RVEA shows the most promising overall performance.

6. Conclusion and Future Works

In this paper, a novel adaptive reference vector adjustment strategy is developed to adjust the predefined reference vectors according to the distribution of current promising solutions. Then, an improved APD method is shown to pose more selection pressure in terms of convergence performance to the candidate population. On the basis of the above two methods, an improved version of RVEA called AP-RVEA is proposed to maintain sound population diversity and converge efficiently to the true PF. The proposed AP-RVEA is compared with five state-of-the-art decomposition-based algorithms on 36 test instances taken from MaF test suite. The HV results and IGD results give the conclusion that AP-RVEA achieves the most competitive performance among the six algorithms.

In the future, we hope that AP-RVEA can be used to solve some real-world problems with irregular problems instead of being limited to test problems. Besides, it may also be meaningful to extend AP-RVEA to solve some constraint problems with irregular PFs [41].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: part II," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[2] C.-H. Chen and J.-H. Chou, "Multiobjective optimization of airline crew roster recovery problems under disruption conditions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 133–144, 2017.

[3] A. S. Sayyad, T. Menzies, and H. Ammar, "On the value of user preferences in search-based software engineering: a case study in software product lines," in *Proceedings of the 2013*

35th International Conference on Software Engineering (ICSE), pp. 492–501, San Francisco, CA, USA, May 2013.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] Q. Hui Li and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[6] W. Qiu, J. Zhu, G. Wu, H. Chen, W. Pedrycz, and P. N. Suganthan, "Ensemble many-objective optimization algorithm based on voting mechanism," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, pp. 1–15, 2020.

[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with Box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[8] X. Cai, H. Sun, Q. Zhang, and Y. Huang, "A grid weighted sum pareto local search for combinatorial multi and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3586–3598, 2019.

[9] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for moea/d," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

[10] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 564–577, 2018.

[11] X. Cai, C. Xia, Q. Zhang et al., "The collaborative local search based on dynamic-constrained decomposition with grids for combinatorial multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–12, 2019.

[12] X. Xinye Cai, Y. Yexing Li, Z. Zhun Fan, and Q. Qingfu Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

[13] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.

[14] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[15] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2018.

[16] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[17] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[18] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[19] H. Chen, G. Wu, W. Pedrycz, P. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1–16, 2019.

[20] F. Wang, Y. Li, F. Liao, and H. Yan, "An ensemble learning based prediction strategy for dynamic multi-objective optimization," *Applied Soft Computing*, vol. 96, Article ID 106592, 2020.

[21] S. Jiang, J. Zhang, and Y. S. Ong, "Asymmetric pareto-adaptive scheme for multiobjective optimization," in *Advances in Artificial Intelligence*, D. Wang and M. Reynolds, Eds., vol. 7106, Springer, Berlin, Germany, 2011.

[22] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[23] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.

[24] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, 2013.

[25] S. Jiang, X. He, and Y. Zhou, "Many-objective evolutionary algorithm based on adaptive weighted decomposition," *Applied Soft Computing*, vol. 84, Article ID 105731, 2019.

[26] Z. Liang, W. Hou, X. Huang, and Z. Zhu, "Two new reference vector adaptation strategies for many-objective evolutionary algorithms," *Information Sciences*, vol. 483, pp. 332–349, 2019.

[27] X. Cai, Z. Mei, and Z. Fan, "A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2335–2348, 2018.

[28] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2018.

[29] H. Zhao, C. Zhang, and B. Zhang, "A decomposition-based many-objective ant colony optimization algorithm with adaptive reference points," *Information Sciences*, vol. 540, pp. 435–448, 2020.

[30] C. Zhou, G. Dai, C. Zhang, X. Li, and K. Ma, "Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems," *Information Sciences*, vol. 465, pp. 232–247, 2018.

[31] F. Wang, Y. Li, H. Zhang, T. Hu, and X.-L. Shen, "An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 220–233, 2019.

[32] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[33] Q. Liu, Y. Jin, M. Heiderich, and T. Rodemann, "Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular pareto fronts," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1726–1733, Wellington, New Zealand, June 2019.

[34] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, and G. Yu, "An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[35] W. Qiu, J. Zhu, G. Wu, M. Fan, and P. N. Suganthan, "Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100776, 2021.

[36] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "A new two-stage evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 748–761, 2019.

[37] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[38] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[39] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[40] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[41] Z. Fan, W. Li, X. Cai et al., "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.

Hindawi
Complexity
Volume 2021, Article ID 6694695, 17 pages
https://doi.org/10.1155/2021/6694695

Research Article

A Novel BBO Algorithm Based on Local Search and Nonuniform Variation for Iris Classification

Lisheng Wei[1,2], **Ning Wang**[1,3] and **Huacai Lu**[1]

[1]*Anhui Key Laboratory of Electric Drive and Control, Anhui Polytechnic University, Wuhu 241000, China*
[2]*Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment of Ministry of Education, AnHui Polytechnic University, Wuhu 241000, China*
[3]*Anhui Key Laboratory of Detection Technology and Energy Saving Devices, Anhui Polytechnic University, Wuhu 241000, China*

Correspondence should be addressed to Lisheng Wei; lshwei_11@163.com and Ning Wang; 18375318758@163.com

Received 6 November 2020; Revised 8 January 2021; Accepted 27 March 2021; Published 15 April 2021

Academic Editor: Shi Cheng

In order to improve the iris classification rate, a novel biogeography-based optimization algorithm (NBBO) based on local search and nonuniform variation was proposed in this paper. Firstly, the linear migration model was replaced by a hyperbolic cotangent model which was closer to the natural law. And, the local search strategy was added to traditional BBO algorithm migration operation to enhance the global search ability of the algorithm. Then, the nonuniform variation was introduced to enhance the algorithm in the later iteration. The algorithm could achieve a stronger iris classifier by lifting weaker similarity classifiers during the training stage. On this base, the convergence condition of NBBO was proposed by using the Markov chain strategy. Finally, simulation results were given to demonstrate the effectiveness and efficiency of the proposed iris classification method.

1. Introduction

Iris classification is especially suitable for recognition with uniqueness, stability, inviolability, and reliability characteristics and has been one of the hottest biological characteristic recognition research spot recently [1]. It has been widely used technology in national defence, financial industry, and entrance guard system [2, 3]. Alfred Wallace and Charles Darwin proposed the theory of biogeography to study the distribution, migration, and extinction of habitats of biological species in the 19th century. Inspired by biogeography, Simon proposed the BBO algorithm in 2008 [4]. BBO is a new swarm intelligence optimization algorithm and solves the optimization problem by simulating the mathematical model of species migration in biogeography. In general, BBO can find a good initial candidate solution and generate an acceptable optimization solution, which has been widely used in image processing [5], scheduling optimization [6, 7], parameter estimation [8], power flow calculation [9], and load analysis [10]. However, BBO has some shortcomings, such as weak search ability and easy to fall into local optimum in the later period of operation. How to use BBO global search ability to improve the iris classification rate has important theoretical and practical value.

In recent years, many researchers have worked on the improvement of the BBO performance. For instance, Wang et al. [11] combined the chaotic mapping strategy with the BBO optimal migration model and proposed a biogeography optimization algorithm based on based on the adaptive population migration mechanism. Feng et al. [12] proposed an improved BBO with random ring topology and Powell's method, in which the local loop topology was used instead of the global topology. The adaptive Powell method was modified to adapt the evolutionary process to improve the accuracy of the solution. Li et al. [13] designed a hybrid algorithm by combining the artificial bee colony algorithm (ABC) with BBO to obtain ABC's exploring ability and BBO's developing ability. Chen et al. [14] combined Cuckoo Search (CS) and bBBO using two search strategies: heterogeneous rhododendron search and biogeography-based discovery. Zhao et al. [15] proposed an optimization method based on two-stage differential biogeography to solve the

problem of premature convergence and reduce rotation variance.

In order to improve the iris classification rate, the NBBO based on local search and nonuniform variation was proposed in this paper. Firstly, the hyperbolic cotangent migration model was used to replace the original linear migration model to improve the adaptability of the algorithm. Then, the local search strategy was added to the BBO migration operation in order to enhance the algorithm search ability and improve the convergence speed of the algorithm. The nonuniform variation was used in the operation to increase the search ability of the algorithm and prevent the local optimum, which could achieve a stronger iris classifier by lifting weaker similarity classifiers during the training stage.

The rest of the paper was organized as follows. Section 2 presented the improved biogeography-based optimization algorithm, including migration model, migration operations, and nonuniform variation operation. Section 3 presented NBBO test on different test functions and iris classification samples. Finally, a brief conclusion and future work are given in Section 4.

2. Methods

In nature, biological species live in different areas with obvious boundaries, which are called habitats. Habitat suitability index (HSI) is used to describe the survival degree of species suitable for habitat. HSI is affected by humidity, temperature, habitat area, and vegetation diversity. These factors are called suitability variables (SIV). Habitats with higher HSI can accommodate more biological species, whereas habitats with lower HSI can only accommodate fewer biological species. BBO is a population-based algorithm, which regards each solution as a habitat, fitness as HSI, and each component of solution as a SIV. The optimization problem can be solved by simulating the migration and mutation process in biogeography. The two main operations of BBO are migration operation and mutation operation [16, 17].

Like other intelligent algorithms, BBO is prone to oscillation in the iterative process, resulting in slow and premature convergence, poor local search ability, and inaccuracy. This section presents the NBBO algorithm in terms of migration model and operation. The flow chart of the improved algorithm is shown in Figure 1.

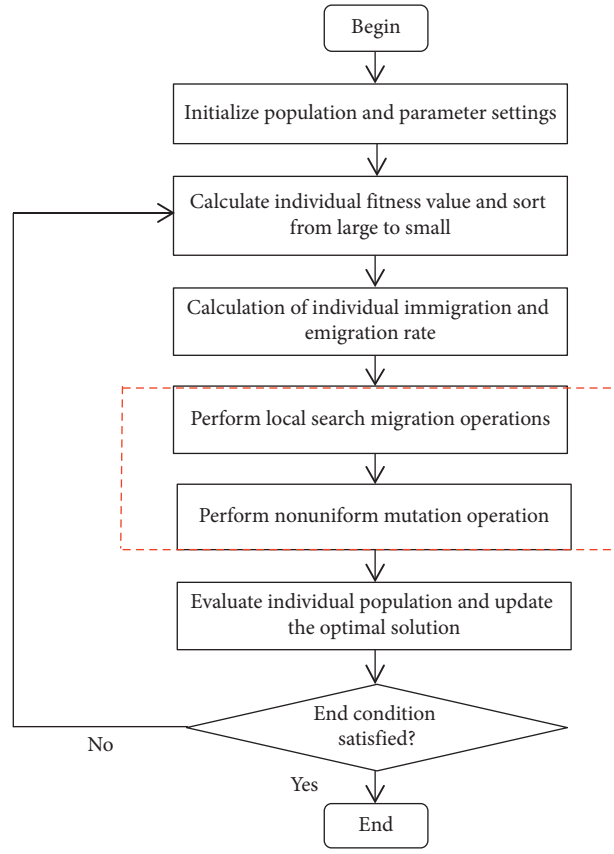


Figure 1: NBBO algorithm flowchart.

2.1. Migration Model. There are different migration models in biogeography. As shown in formula (1), the original BBO adopts the simplest linear migration model. But, the ecosystem is nonlinear in nature; small changes in a certain part of the system may have complex effects on the whole system. For the actual species, the migration process is more complicated than what is described by the linear model. Among the related improved algorithms, the cosine mobility model proposed by Ma performs best under most of the functions tested [18], and the mobility model close to the natural law is better than the simple linear model. In order to better adapt to the nonlinear migration problem, this paper proposes a hyperbolic cotangent nonlinear migration model to improve the performance of the basic BBO algorithm.

$$\begin{cases} \lambda_k = I \cdot \left(1 - \dfrac{k}{n}\right), \\ \mu_k = E \cdot \dfrac{k}{n}, \end{cases} \tag{1}$$

$$\lambda_k = \frac{I}{2} \cdot \left(-\frac{e^{k\pi/n} - e^{-k\pi/n}}{e^{k\pi/n} + e^{-k\pi/n}} + 1\right), \tag{2}$$

$$\mu_k = \frac{E}{2} \cdot \left(\frac{e^{k\pi/n} - e^{-k\pi/n}}{e^{k\pi/n} + e^{-k\pi/n}} + 1\right), \tag{3}$$

where λ_k and μ_k are related to the number of species k and n is the maximum number of species. λ_k represents the immigration rate which refers to the probability of species moving into the habitat, μ_k represents the emigration rate which refers to the probability of species moving out of the habitat, I represents the maximum immigration rate, and E represents the maximum emigration rate, all of which are functions of the number of habitat populations k. When the number of habitat species is 0, the immigration rate is the highest, that is, $\lambda_k = I$, the migration rate is the smallest, that is, $\mu_k = 0$. When the number of species reaches the

maximum number of species n that can be accommodated in the habitat, the immigration rate λ_k drops to 0, and the migration rate reaches the maximum, that is, $\mu_k = E$. When the number of species is k_0, the immigration rate and the emigration rate of the species are equal, that is, $\lambda_k = \mu_k$, and the dynamic equilibrium state is reached at this time.

In the hyperbolic tangential mobility model, the trend of mobility changing with species number is like that of cosine model, but the amplitude is more moderate than that of cosine model.

2.2. Local Search Migration Operations. In the original BBO, a generalized search was carried out based on the immigration rate and the emigration rate, a habitat with a higher emigration rate was randomly selected, and the characteristics of the corresponding position (SIV) were moved to the habitat with low HSI. However, in some habitats with high migration rates, the characteristics of the corresponding locations SIV are not necessarily the best for the migrated habitats. If migrated directly, it may result in a decrease in the suitability index of the move into habitat.

Therefore, in order to enhance the global search ability of the algorithm and improve the convergence speed of the algorithm, the selected migration characteristics can be modified by using equation (6), which can reduce the poor migration characteristics from the habitat with higher migration rate to the low migration rate. Similarly, if you move out of the same habitat with a high migration rate each time, as the number of evolutions increases, the population diversity decreases, and the convergence speed will slow down:

$$C = \text{MinVal} + \text{rand} * (\text{MaxVal} - \text{MinVal}), \quad (4)$$

$$\text{randTemp} = \text{rand}(1, 2), \quad (5)$$

$$X'(k, j) = X(k, j) + C * \text{randTemp} * X(k, j), \quad (6)$$

where C represents a space allocated to each individual in the population for storing algorithm parameters. $X(k, j)$ represents the current individual.$X'(k, j)$ represents the new individual generated by the local search. MinVal = −1 and MaxVal = 1.

2.3. Nonuniform Variation Operation. In the early stage of iteration, the population position is usually far from the optimal solution and needs to be iterated many times to find a feasible search space when the search radius is too small. At the end of iteration, the population position is close to or near the optimal solution. A small range of searches is used as a fine-tuning of the solution vector. It is shown by equation (7), as the traditional mutation operator cannot search the optimal solution efficiently, the nonuniform mutation operator is introduced. The whole population is searched at a large range during the early stage of search. As the number of iterations increases, the search range is gradually narrowed, and the algorithm is effectively balanced to escape from local minimum.

$$m_s = m_{\max} \cdot \left(1 - \frac{P_s}{P_{\max}}\right), \quad (7)$$

where P_s denotes the probability for the S number of species; S denotes the species number; and $m_{\max}$ denotes the maximum variation rate given; $P_{\max} = \max(P_s)$.

The specific operation of nonuniform variation is like uniform variation, but it focuses on searching for small areas near the original individual. Assuming that the variation of the k_{th} component X_k of the t generation (t is the current number of iterations) of Habitat $X_i^t = \{X_1, X_2, \ldots, X_k, \ldots, X_n\}$ is performed, the variation component is as follows:

$$X_k' = X_k + \Delta(t, U_{\max} - X_k), \quad \text{if rand} < 0.5, \quad (8)$$

$$X_k' = X_k - \Delta(t, X_k - U_{\min}), \quad \text{if rand} \geq 0.5, \quad (9)$$

where $U_{\max}$ and $U_{\min}$ are the upper and lower limits of component X_k, respectively. $U_{\max} - X_k$ and $X_k - U_{\min}$ represent random numbers in the range of $[0, y]$ and meet the nonuniform distribution:

$$\lim_{t \longrightarrow \infty} \Delta(t, y) = 0. \quad (10)$$

This feature allows the mutation operator to search the entire space evenly in the initial stage of the algorithm (when t is very small), and the latter part of the algorithm focuses on several local range exact searches. $\Delta(t, y)$ can be defined as follows:

$$\Delta(t, y) = y \cdot \left(1 - r^{(1-(t/T))\cdot b}\right), \quad (11)$$

where r is a random number that fits evenly within the $[0, 1]$ range, T is the maximum number of iterations, and b is the system parameter that determines the degree of dependency on the number of iterations, select here $b = 2$.

Then, the specific steps of NBBO algorithm are as follows: (Algorithm 1)

2.4. Convergence Analysis. By using Markov chain strategy, we can analyze the convergence of the proposed NBBO algorithms [19, 20]. Let $X(t) = \{x_1(t), x_2(t), \ldots, x_n(t)\}$ be a population with improved algorithm algebra of t. We divide $X(t)$ into n subsets based on the fitness of the population. $x_i(t)$ $(i = 1, 2, \ldots, n)$ is a subset of them; next, we use Markov chain to analyze the global convergence of the improved BBO algorithm.

The evolution process of the NBBO algorithm is mainly composed of selection, migration and mutation operations, which are independent of evolution algebra. Considering the number of individuals of BBO algorithm, the evolution process of the NBBO algorithm satisfies the finite homogeneous Markov chain. Population subset $x_i(t)$ $(i = 1, 2, \ldots, n)$ can be equivalent to a state on a finite homogeneous Markov chain, p_i means the probability of being in state $x_i(t)$, and then $p_i > 0$ and $\sum_{i=1}^{n} p_i = 1$.

The transition probability $p_{ij} = p(x_i \longrightarrow x_j)$ of state $x_i(t)$ is as follows:

$$\begin{cases} p_{ij} = 0, & i < j \\ p_{ij} > 0, & i \geq j. \end{cases} \tag{12}$$

Then, the state transition matrix P of the Markov chain is

$$P = \begin{bmatrix} p_{11} & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} = \begin{bmatrix} p_{11} & 0 \\ Q_2 & Q_1 \end{bmatrix}, \tag{13}$$

where

$$\begin{aligned} Q_1 &= \begin{bmatrix} p_{22} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ p_{n2} & \cdots & p_{nn} \end{bmatrix}, \\ Q_2 &= (p_{21}, p_{31}, \cdots, p_{n1})^T. \end{aligned} \tag{14}$$

It can be found by formulas (12) and (13) $Q_2 \neq 0, Q_1 \neq 0$. Since the sum of the probabilities of each row in the Markov transition matrix is 1, $p_{11} = 1 \neq 0$. It can be concluded that the state transition matrix P of Markov chain satisfies the reducible random matrix. According to [21], the definition of a reductive random matrix is as follows:

Let matrix Z of order N be a reducible random matrix which can be obtained by the same row transformation and column transformation, namely,

$$Z = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix}, \tag{15}$$

where C is the m ($m \leq N$) order primitive matrix and R and T are nonzero matrices.

Then, we have

$$Z^{\infty} = \lim_{k \longrightarrow \infty} Z^k = \lim_{k \longrightarrow \infty} \begin{bmatrix} C^k & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{bmatrix}. \tag{16}$$

The above matrix is a stable random matrix, and Z^{∞} satisfies the following conditions:

$$Z^{\infty} = \left[Z_{ij}\right]_{N*N}, \begin{cases} Z_{ij} > 0, & 1 \leq i \leq N, 1 \leq j \leq m, \\ Z_{ij} = 0, & 1 \leq i \leq N, m < j \leq N. \end{cases} \tag{17}$$

Therefore, state transition matrix P is a reductive random matrix, then

$$\begin{aligned} P^{\infty} &= \lim_{t \longrightarrow \infty} \begin{bmatrix} p_{11} & 0 \\ Q_2 & Q_1 \end{bmatrix}^t \\ &= \lim_{t \longrightarrow \infty} \begin{bmatrix} p_{11}^t & 0 \\ \sum_{i=1}^{t-1} Q_1^i Q_2 p_{11}^{t-i} & Q_1^t \end{bmatrix} \\ &= \lim_{t \longrightarrow \infty} \begin{bmatrix} p_{11}^{\infty} & 0 \\ Q_2^{\infty} & 0 \end{bmatrix}. \end{aligned} \tag{18}$$

In formula (12) $p_{11}^{\infty} = p_{11} = 1$ and $Q_1^{\infty} = 0$, since the sum of the probabilities of each line in the Markov transition matrix is 1; there must be $Q_2^{\infty} = (1, 1, \ldots, 1)^T$, then

$$P^{\infty} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{19}$$

Formula (19) shows that when the number of iterations $t \longrightarrow \infty$, probability $p_{i1}^{\infty} = 1$ ($i = 1, 2, \ldots, n$), That is, no matter how the initial state can finally converge to the global optimal solution with probability 1, it can be concluded that

$$\lim_{t \longrightarrow \infty} p\{Y_t = f^*\} = 1, \tag{20}$$

where Y_t is the optimal fitness value of population $X(t)$, f^* represents the global optimal fitness value, and $p\{Y_t = f^*\}$ indicates the probability that the optimal individual of the t generation is the global optimal value. It is proved that the NBBO algorithm has global convergence.

3. Simulation Experiments

3.1. Test Function Simulation. In order to test the optimization performance of the proposed IMBBO algorithm, 13 standard test functions were selected for simulation experiments [22, 23]. Table 1 shows the test function expression, the range of values for each independent variable, and the theoretical optimal value. We select the optimal value, average value, and standard deviation of the test results as the evaluation criteria to test the algorithm performance.

The experimental function of Table 1 is numerically tested using the proposed NBBO algorithm proposed in this paper, and the basic BBO algorithm, grey wolf optimization algorithm [24] (DE), elephant herding optimization [25] (EHO), salp swarm algorithm [26] (DEBBO), and particle swarm optimization biogeography optimization algorithm [27] and (BBOPSO) for comparison of results. The experimental parameter settings of the algorithm are shown in Table 2.

To study the performance of the algorithm in different dimensions, the dimensions of test function $f_1 - f_3$ and $f_5 - f_7$ are $D = 10, 30, 50$, the dimension of test functions f_4 is $D = 2$, and the dimension of test functions f_8 is $D = 4$. In order to ensure that the comparison between each method is fair, we ensure that the experimental environment of each method is consistent during the experiment: operating system: window 10, memory: 8 GB, programming language: MATLAB 2016, and ensure that the setting of common parameters of the algorithm is consistent, and the parameters of different algorithms are set according to the parameters of specific references. As the algorithm is running randomly, the six algorithms independently run each test function 30 times to avoid unnecessary errors caused by the random operation of the algorithm.

Tables 3 and 4 show the recorded optimal function value, average value, and standard deviation obtained by the

```
Begin
Algorithmic parameters setting:
Population size = 50,
Maximum number of iterations G = 500,
Maximum immigration rate I = 1,
Maximum emigration rate E = 1,
Mutation probability P_m = 0.05,
Number of elites retained Keep = 2.
    /* Initialization */
Random generation of a set of initial habitats constitutes the initial population.
Calculate the fitness value of each habitat in the population.
While not T /* T is the condition for the end of the iteration */
Sort the habitat in descending order according to the fitness value.
The immigration rate λ_i and emigration rate μ_i of each habitat were calculated according to formula (1).
    /* End of initialization */
    /* Migration */
p_i
  Select H_i according to the immigration probability λ_i
  for j = 1: D /* D is dimension */
    if rand < λ_i
      Select H_j according to the migration probability μ_i
        Randomly select a characteristic variable SIV from H_j
        Replace a random characteristic variable SIV in H_i with SIV
    else
        Perform a local search on the characteristic variable SIV
          according to formula (6) can get SIV'.
        Replace random characteristic variable SIV in H_i with SIV'.
    end if
  end for
end for
    /* End of migration */
    /* Mutation */
for i = 1: Size
  Calculate the probability of mutation P_s according to λ_i and μ_i
  Selection of unmutated habitat H_i based on P_s
  if rand(0, 1) < m_s then
    if rand < 0.5
    Replace H_i(SIV) with a randomly generated SIV by formula (8)
      if rand ≥ 0.5
        Replace H_i(SIV) with a randomly generated SIV by (9)
      end if
    end if
  end if
end for
/* End of mutation */
          Recalculate the habitat fitness values
end while
end
```

Algorithm 1: The main procedure of NBBO.

algorithm. In the iterative process of the algorithm, we record the best fitness of each evaluation, find the average optimal value of each evaluation and draw a curve, which reflects the convergence trend of the algorithm. The convergence curves of some test functions are shown in Figures 2 and 3.

In order to compare the optimization performance of each algorithm more easily and clearly, the algorithm with the best optimization performance of the optimal value, average value and standard deviation of each test function is boldly marked in Tables 3 and 4. From Tables 3 and 4 and the simulation diagram, we find that the convergence speed of the analytical algorithm can be qualitatively and quantitatively analyzed. For the simple multidimensional unimodal function in Figure 2, the NBBO algorithm has faster convergence speed and solution precision in the later iteration although its advantage is not obvious in the initial stage of iteration. For the more complex multidimensional multimodal function in Figure 3, the other five algorithms have the phenomenon of search stagnation or slow search, i.e.,

Table 1: Test function.

Function	Expression	Ranges	Theory optimal
$f1$: Quartic	$f(x_1) = \sum_{i=1}^{n} ix^4 + \text{random}[0, 1)$	(−1.28, 1.28)	0
$f2$: Sphere	$f(x_4) = \sum_{i=1}^{n} x_i^2$	{−100, 100)	0
$f3$: Powell	$f(x_5) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	(−4, 5)	0
$f4$: Easom	$f(x_{10}) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	(−100, 100)	−1
$f5$: Alpine	$f(x_6) = \sum_{i=1}^{n} \lvert x_i \sin x_i + 0.1x_i \rvert$	(10, 10)	0
$f6$: Griewank	$f(x_7) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	(−600, 600)	0
$f7$: Ackley	$f(x_8) = -20\exp(-0.2\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}) - \exp((1/n)\sum_{i=1}^{n} \cos 2\pi x_i) + 20 + e$	(−32, 32)	0
$f8$: Kowalik	$f(x_{13}) = \sum_{i=1}^{11} [a_i - ((x_1(b_i^2 + b_i x_2))/(b_i^2 + b_i x_3 + x_4))]^2$	(−5, 5)	$3.075E-04$

Table 2: Algorithm parameter settings.

Setting object	Parameter	Value
Common part	Population size	Size = 50
	The maximum number of iterations	$G = 500$
	Elite reserves	Keep = 2
BBO/BBOPSO	Habitat modification probability	$P_{\text{mod}} = 1$
	Maximum immigration	$I = 1$
	Maximum migration	$E = 1$
	Mutation probability	$P_m = 0.05$
DE	Weighting factor	$F = 0.5$
	Crossover probability	CR = 0.5
EHO	The scale factors	$\alpha = 0.5$, $\beta = 0.5$ nClan = 5
	The number of clans	
DEBBO	Weighting factor	$F = 0.5$
	Crossover probability	CR = 0.9
	DE mutation scheme	DE/rand/1/bin
The proposed NBBO	Habitat modification probability	$P_{\text{mod}} = 1$
	Maximum immigration	$I = 1$
	Maximum migration	$E = 1$
	Mutation probability	$P_m = 0.05$
	System parameters	$b = 2$

they fall into local optimum and are difficult to jump out. Then, by comparing the optimal value of each test function and the average value of the optimization results, we can see that IMBBO algorithm has achieved a smaller value. Although the optimization results of DE, DEBBO, and BBOPSO algorithms also reach the global optimal value in the optimization of some test functions with different dimensions, it can be seen from the optimization results of single peak function and multi peak function that NBBO algorithm has a great advantage in the optimization ability. Figures 2 and 3 show that the proposed NBBO algorithm can not only improve the convergence of the algorithm but also improve the evolution ability and the ability to jump out of the local optimal solution. It is superior to the other four comparison algorithms in the stability of the optimization result. In summary, when solving the optimization problem of single-peak or multi-peak functions, the proposed NBBO can quickly and effectively find the optimal solution under a short number of iterations, and its optimal solution has higher precision and stability.

Considering the fixed dimension test function has few dimensions to be adjusted, the accuracy and stability of each algorithm are relatively high. For the multidimensional test function, when its dimension D is increased from 10 to 30 or 50, the accuracy and stability of the algorithm have decreased. This is because the objective function is more complicated and the algorithm needs to make more adjustments. However, no matter how the dimension changes, the proposed NBBO algorithm still maintains a stable and high-precision optimization performance for high-dimensional complex functions. In reference [28], derrick proposed that for the evaluation of the performance of the improved evolutionary algorithm, only based on the average value and the standard deviation value to compare the advantages and disadvantages of the algorithm is not rigorous enough, so statistical test should be carried out. The statistical test shows that the improved algorithm has significant improvement advantages over other existing algorithms. Therefore, in order to verify the significant difference between the IMBBO algorithm and other algorithms, a nonparametric estimation method, Wilcoxon rank sum test, is used for statistical analysis. For each test function, the 30 times solution results of IMBBO algorithm are tested with those of BBO, DE, EHO, DEBBO, and BBOPSO algorithms respectively, and H0 is assumed: the solution results of the two algorithms are generally the same; H1: the solution

Table 3: Unimodal function test results.

Function	Dimension	Evaluating indicator	Reference [4]BBO	Reference [24]DE	Reference [25]EHO	Reference [26]DEBBO	Reference [27]BBOPSO	The proposed method
$f_1(x)$ Quartic	$D=10$	Best	$3.40E-03$	$1.70E-03$	$2.10E-03$	$1.81E-04$	$7.92E-05$	**5.78 E – 05**
		Ave	$1.37E-02$	$4.10E-03$	$6.60E-03$	$9.45E-04$	$1.20E-03$	**4.51 E – 04**
		Std	$9.80E-03$	$1.70E-03$	$3.70E-03$	$6.91E-04$	$8.62E-04$	**3.25 E – 04**
	$D=30$	Best	$7.58E-02$	$3.50E-02$	$8.44E-02$	$1.05E-02$	$6.20E-03$	**1.10 E – 03**
		Ave	$1.47E-01$	$5.46E-02$	$1.33E-01$	$1.46E-02$	$2.23E-02$	**2.70 E – 03**
		Std	$4.41E-02$	$1.03E-02$	$4.13E-02$	$4.20E-03$	$9.10E-03$	**2.00 E – 03**
	$D=50$	Best	$2.51E-01$	$1.21E-01$	$1.98E-01$	$3.47E-02$	$4.63E-02$	**3.00 E – 03**
		Ave	$5.67E-01$	$2.29E-01$	$5.37E-01$	$5.61E-02$	$7.82E-02$	**8.50 E – 03**
		Std	$2.08E-01$	$5.16E-02$	$1.38E-01$	$1.56E-02$	$2.22E-02$	**4.20 E – 03**
$f_2(x)$ Sphere	$D=10$	Best	$1.94E+00$	$1.18E-20$	$2.35E-04$	$6.72E-32$	$2.02E-28$	**3.66 E – 45**
		Ave	$8.38E+00$	$3.52E-20$	$3.74E-04$	$1.69E-29$	$1.37E-27$	**5.79 E – 43**
		Std	$6.12E+00$	$2.77E-20$	$6.88E-05$	$2.06E-29$	$1.49E-27$	**9.78 E – 43**
	$D=30$	Best	$7.13E+01$	$1.20E-03$	$3.00E-03$	$1.40E-10$	$6.73E-07$	**2.09 E – 24**
		Ave	$1.43E+02$	$2.60E-03$	$3.60E-03$	$5.83E-10$	$1.24E-05$	**1.43 E – 22**
		Std	$4.64E+01$	$2.20E-03$	$0.31E-04$	$4.90E-10$	$3.35E-05$	**3.91 E – 22**
	$D=50$	Best	$5.93E+02$	$1.58E+01$	$7.90E-03$	$2.38E-05$	$4.11E-02$	**4.66 E – 18**
		Ave	$7.27E+02$	$2.30E+01$	$9.00E-03$	$1.62E-04$	$1.64E-01$	**6.83 E – 17**
		Std	$1.34E+02$	$4.63E+00$	$1.20E-03$	$1.54E-04$	$9.32E-02$	**8.48 E – 17**
$f_3(x)$ Powell	$D=10$	Best	$2.47E-02$	$2.04E-05$	$4.64E-06$	$3.89E-04$	$2.63E-04$	**8.88 E – 13**
		Ave	$3.01E-01$	$1.52E-04$	$7.51E-06$	$2.40E-03$	$1.45E-03$	**5.16 E – 07**
		Std	$2.54E-01$	$1.04E-04$	$2.22E-06$	$1.80E-03$	$1.24E-03$	**7.21 E – 07**
	$D=30$	Best	$2.47E+00$	$6.19E+01$	$2.36E-04$	$1.21E-01$	$5.35E-02$	**1.13 E – 10**
		Ave	$5.77E+00$	$1.36E+01$	$4.10E-04$	$3.92E-01$	$2.90E-01$	**1.26 E – 06**
		Std	$3.75E+00$	$4.70E+01$	$1.50E-04$	$1.61E-01$	$1.25E-01$	**2.10 E – 06**
	$D=50$	Best	$1.73E+01$	$1.04E+03$	$2.20E-03$	$2.02E+00$	$1.63E+00$	**4.07 E – 09**
		Ave	$5.16E+01$	$1.87E+03$	$3.30E-03$	$4.55E+00$	$2.55E+00$	**4.69 E – 06**
		Std	$2.44E+01$	$4.83E+02$	$6.48E-04$	$2.61E+00$	$9.34E-01$	**6.33 E – 09**
$f_4(x)$ Easom	$D=2$	Best	$-9.13E-01$	**−1.00 E + 00**	$-9.95E-01$	**−1.00 E + 00**	**−1.00 E + 00**	**−1.00 E + 00**
		Ave	$-5.00E-01$	**−1.00 E + 00**	$-9.60E-01$	**−1.00 E + 00**	**−1.00 E + 00**	**−1.00 E + 00**
		Std	$4.45E-01$	**0.00 E + 00**	$3.55EE-02$	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**

Table 4: Multimodal function test results.

Function	Dimension	Evaluating indicator	Reference [4] BBO	Reference [24] DE	Reference [25] EHO	Reference [26] DEBBO	Reference [27] BBOPSO	The proposed method
$f_5(x)$ Alpine	$D=10$	Best	$2.59E-02$	$1.41E-19$	$3.59E-06$	$4.91E-06$	$9.43E-09$	**$1.26E-26$**
		Ave	$5.18E-02$	$2.97E-16$	$4.66E-06$	$8.09E-05$	$1.58E-05$	**$8.04E-25$**
		Std	$2.26E-02$	$5.21E-16$	$8.50E-07$	$5.62E-05$	$1.77E-05$	**$1.73E-24$**
	$D=30$	Best	$3.24E-01$	$7.41E+00$	$5.22E-06$	$4.50E-03$	$4.43E-03$	**$9.64E-16$**
		Ave	$7.83E-01$	$1.11E+01$	$7.12E-05$	$6.21E-02$	$9.06E-03$	**$1.05E-13$**
		Std	$2.94E-01$	$1.97E+00$	$1.75E-05$	$5.85E-02$	$3.07E-03$	**$3.65E-13$**
	$D=50$	Best	$2.32E+00$	$3.12E+01$	$1.68E-04$	$3.33E-01$	$4.26E-02$	**$3.55E-13$**
		Ave	$3.26E+00$	$3.62E+01$	$2.29E-04$	$1.15E+00$	$1.58E-01$	**$1.03E-10$**
		Std	$5.16E-01$	$2.45E+00$	$5.69E-05$	$5.32E-01$	$1.41E-01$	**$2.92E-10$**
$f_6(x)$ Griewank	$D=10$	Best	$1.00E+00$	$1.00E+00$	$1.10E+00$	**$0.00E+00$**	**$0.00E+00$**	**$0.00E+00$**
		Ave	$1.07E+00$	$1.00E+00$	$1.14E+00$	$2.48E-02$	$2.95E-02$	**$0.00E+00$**
		Std	$4.12E-02$	**$0.00E+00$**	$2.33E-02$	$1.57E-02$	$2.54E-02$	**$0.00E+00$**
	$D=30$	Best	$1.88E+00$	$1.00E+00$	$2.90E+00$	$6.03E-10$	$3.66E-07$	**$0.00E+00$**
		Ave	$2.40E+00$	$1.00E+00$	$3.61E+00$	$1.50E-03$	$1.24E-02$	**$0.00E+00$**
		Std	$3.36E-01$	$7.20E-04$	$5.98E-01$	$3.20E-03$	$2.40E-02$	**$0.00E+00$**
	$D=50$	Best	$4.86E+00$	$1.12E+00$	$7.72E+00$	$4.22E-05$	$9.11E-02$	**$02.22E-17$**
		Ave	$8.00E+00$	$1.18E+00$	$9.82E+00$	**$1.90E-03$**	$1.78E-01$	
		Std	$2.26E+00$	$3.33E-02$	$1.01E+00$	**$5.00E-03$**	$6.79E-02$	**$7.02E-17$**
$f_7(x)$ Ackley	$D=10$	Best	$8.70E-01$	$3.43E-11$	$7.20E-03$	**$2.66E-15$**	$6.22E-15$	**$2.66E-15$**
		Ave	$2.05E+00$	$8.75E-11$	$8.10E-03$	**$2.66E-15$**	$8.85E-13$	**$2.66E-15$**
		Std	$6.17E-01$	$6.33E-11$	$6.43E-04$	**$0.00E+00$**	$2.54E-12$	**$0.00E+00$**
	$D=30$	Best	$3.58E+00$	$1.28E-02$	$1.44E-02$	$5.12E-06$	$1.08E-04$	**$1.23E-13$**
		Ave	$4.30E+00$	$1.63E-02$	$1.56E-02$	$6.50E-06$	$2.66E-04$	**$1.23E-13$**
		Std	$4.09E-01$	$2.30E-03$	$7.98E-04$	$2.13E-06$	$1.25E-04$	**$1.07E-12$**
	$D=50$	Best	$5.43E+00$	$2.14E+00$	$1.71E-02$	$8.84E-04$	$2.74E-02$	**$1.75E-10$**
		Ave	$6.18E+00$	$2.57E+00$	$1.88E-02$	$1.70E-03$	$7.18E-02$	**$9.85E-10$**
		Std	$5.51E-01$	$2.98E-01$	$9.30E-02$	$1.20E-03$	$6.75E-02$	**$9.57E-10$**
$f_8(x)$ Kowalik	$D=4$	Best	$6.13E-04$	$3.17E-04$	$3.24E-04$	$3.16E-04$	$4.91E-04$	**$3.08E-04$**
		Ave	$2.70E-03$	$6.10E-04$	$3.52E-04$	$9.99E-04$	$7.38E-04$	**$3.23E-04$**
		Std	$4.30E-03$	$4.01E-04$	$2.27E-05$	$1.20E-03$	$3.14E-04$	**$2.26E-05$**

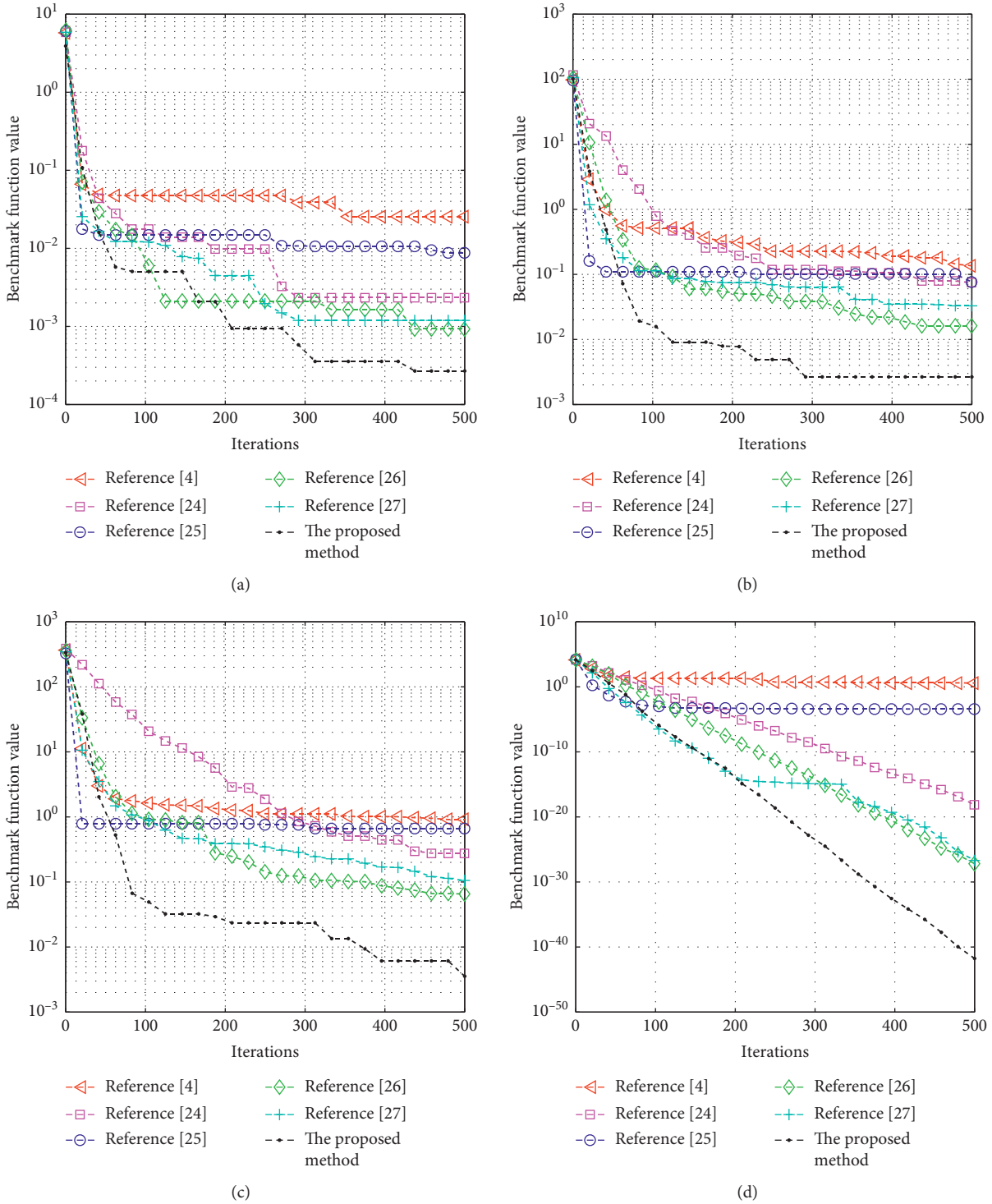


Figure 2: Continued.

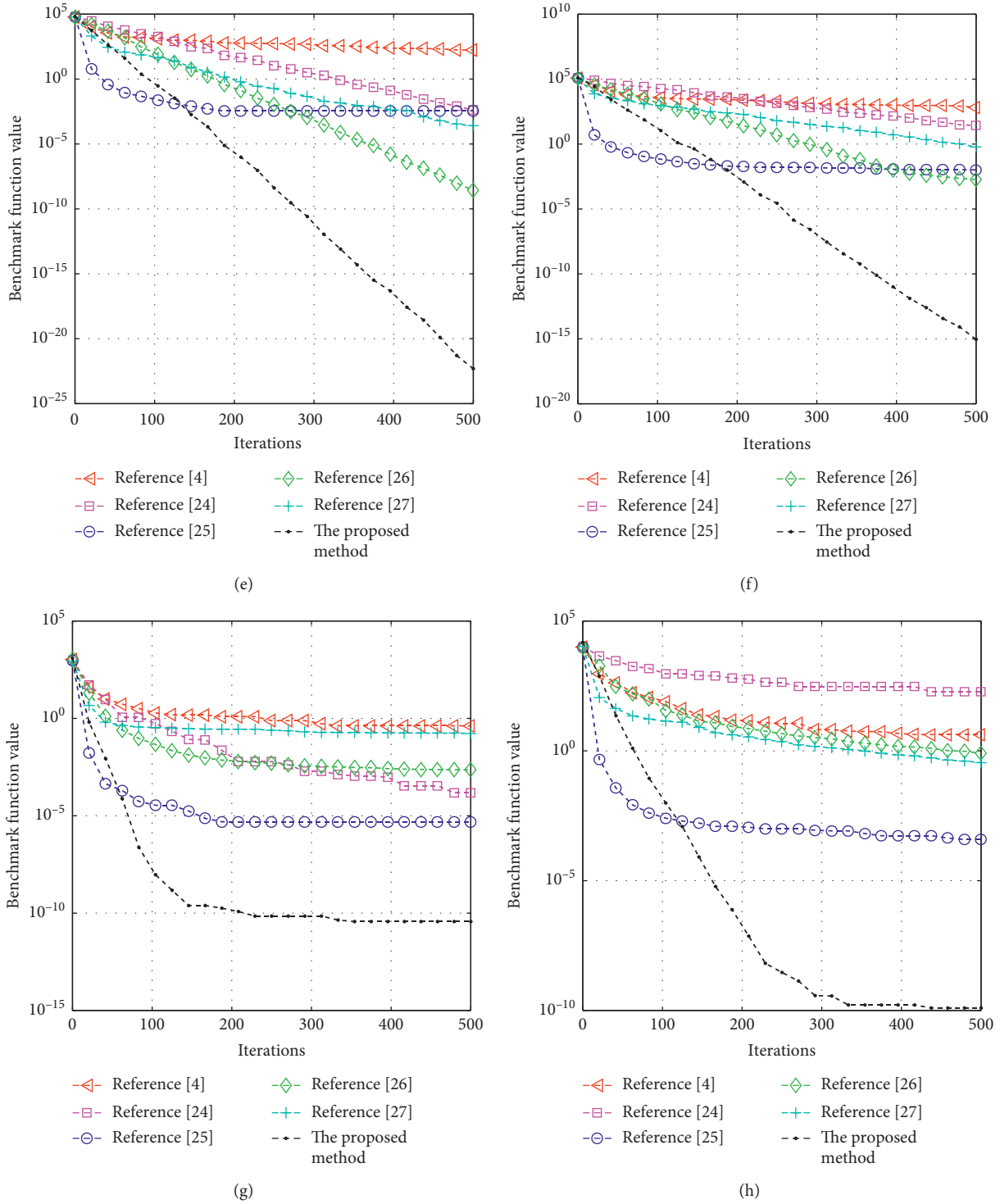


Figure 2: Continued.

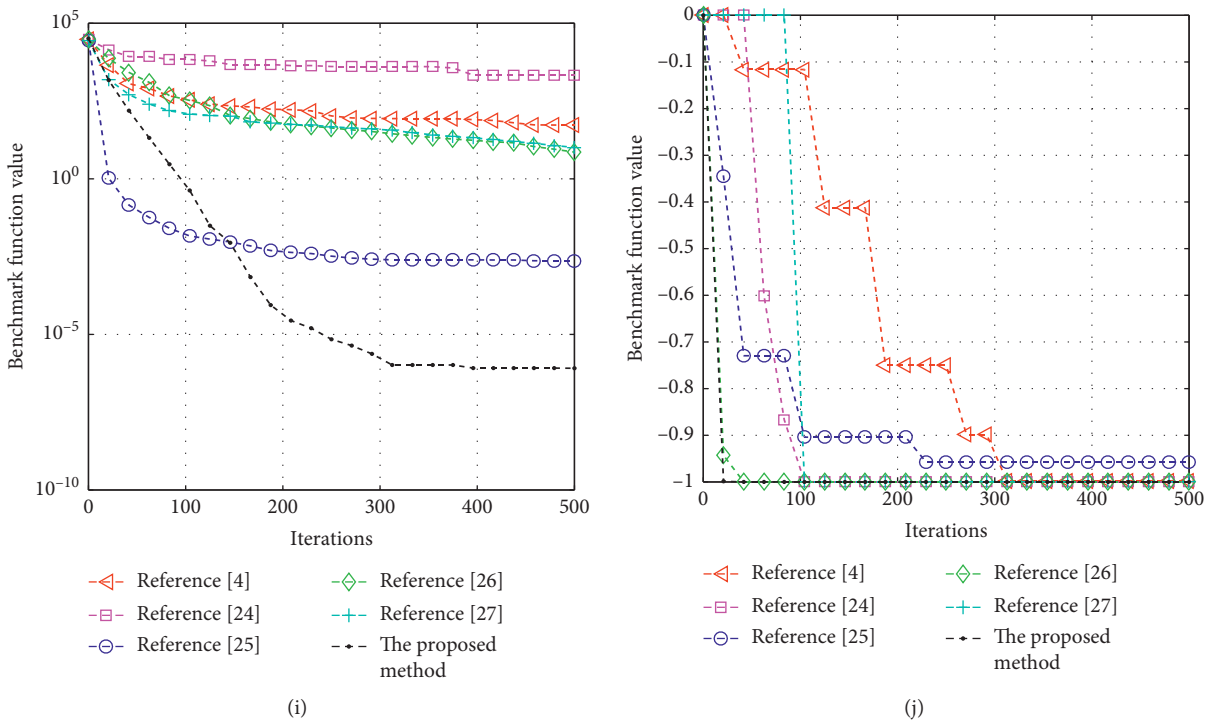


Figure 2: Unimodal function simulation results diagram. (a) f_1: Quartic ($d = 10$), (b) f_1: Quartic ($d = 30$), (c) f_1: Quartic ($d = 50$), (d) f_2: Sphere ($d = 10$), (e) f_2: Sphere ($d = 30$), (f) f_2: Sphere ($d = 50$), (g) f_3: Powell ($d = 10$), (h) f_3: Powell ($d = 30$), (i) f_3: Powell ($d = 50$), (j) f_4: Easom ($d = 2$).

results of the two algorithms are generally different. Set the threshold value of significance level of zero hypothesis test $\alpha = 0.05$. When the test result $P < 0.05$, reject the zero hypothesis; the experimental results of the two algorithms have significant difference; when the test result $P > 0.05$, accept the zero hypothesis; the experimental results of the two algorithms have no significant difference. The specific inspection results are shown in Tables 5 and 6.

Tables 5 and 6 show the test results of test function $f_1 - f_8$. Among them, for the probability p value less than the significance level threshold value of 0.05, bold display; "W" is the significance test judgment result, combined with Tables 3 and 4, use "+" to indicate that NBBO is more accurate and significant than other algorithms, use "−" to indicate that NBBO is less accurate and significant than other algorithms, and use "=" to indicate that NBBO algorithm. There is no significant difference in the accuracy of other algorithms; "N/A" means that significance judgment cannot be made. From the test results, most of the test p values are less than 0.05 and "W" is "+," so zero hypothesis is rejected. Therefore, there are significant differences between the calculation results of IMBBO and BBO, DE, EHO, DEBBO, and BBOPSO, and NBBO is significantly better.

To sum up, the biogeography-based optimization algorithm based on local search and nonuniform variation has good results for most of the benchmark functions, which effectively solves the slow convergence speed of the original BBO algorithm and easy to fall into the local optimization problem in the later stage of the search and greatly improves the convergence speed and search accuracy of the BBO algorithm without adding too many programming steps.

3.2. Iris Classification Based on NBBO Multilayer Perceptron. In order to test the training performance of NBBO multilayer perceptron for iris classification, the mean square error (MSE) of the actual output value of multilayer perceptron and the expected output value is used as an indicator to measure the performance of multilayer perceptron [29]. MSE of all training samples is calculated as follows:

$$\mathrm{MSE} = \frac{1}{s}\sum_{k=1}^{s}\sum_{i=1}^{m}\left(o_i^k - d_i^k\right)^2, \tag{21}$$

where s is the number of training samples, m is the number of outputs, d_i^k is the expected output of the i input unit under k training samples, and o_i^k is the actual output of the i input unit under k training samples. The MSE value is smaller is equal to the performance of the multilayer perceptron is better, and vice versa. The HIS of the i habitat is calculated by

$$\mathrm{HSI}\left(\mathrm{Habitat}_i\right) = E\left(\mathrm{Habitat}_i\right). \tag{22}$$

Figure 4 shows the schematic diagram of a multilayer perceptron based on NBBO algorithm. NBBO algorithm receives the average MSE of all training samples and all expected output samples as the objective function for iterative training and adjusts the link weights and deviations

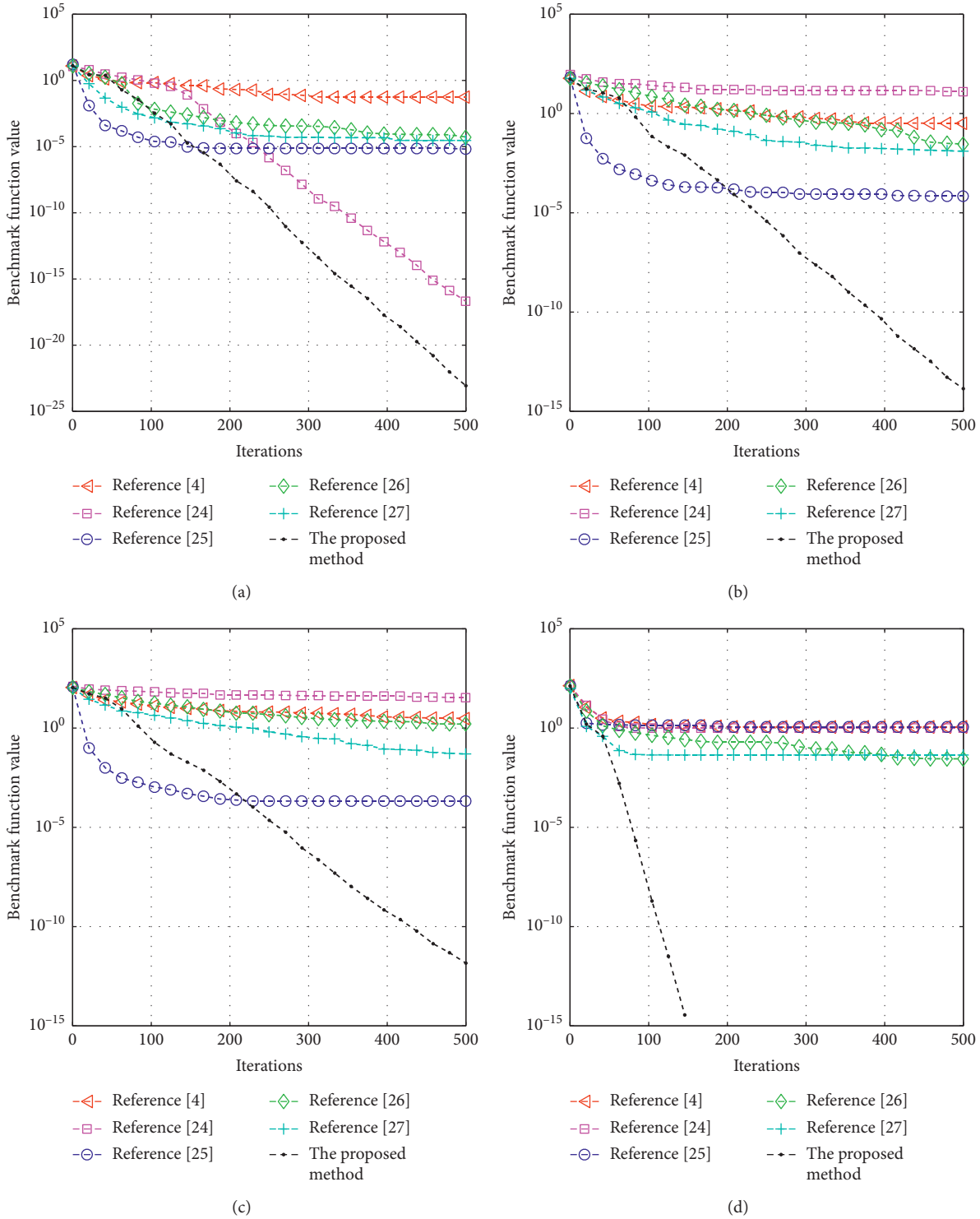
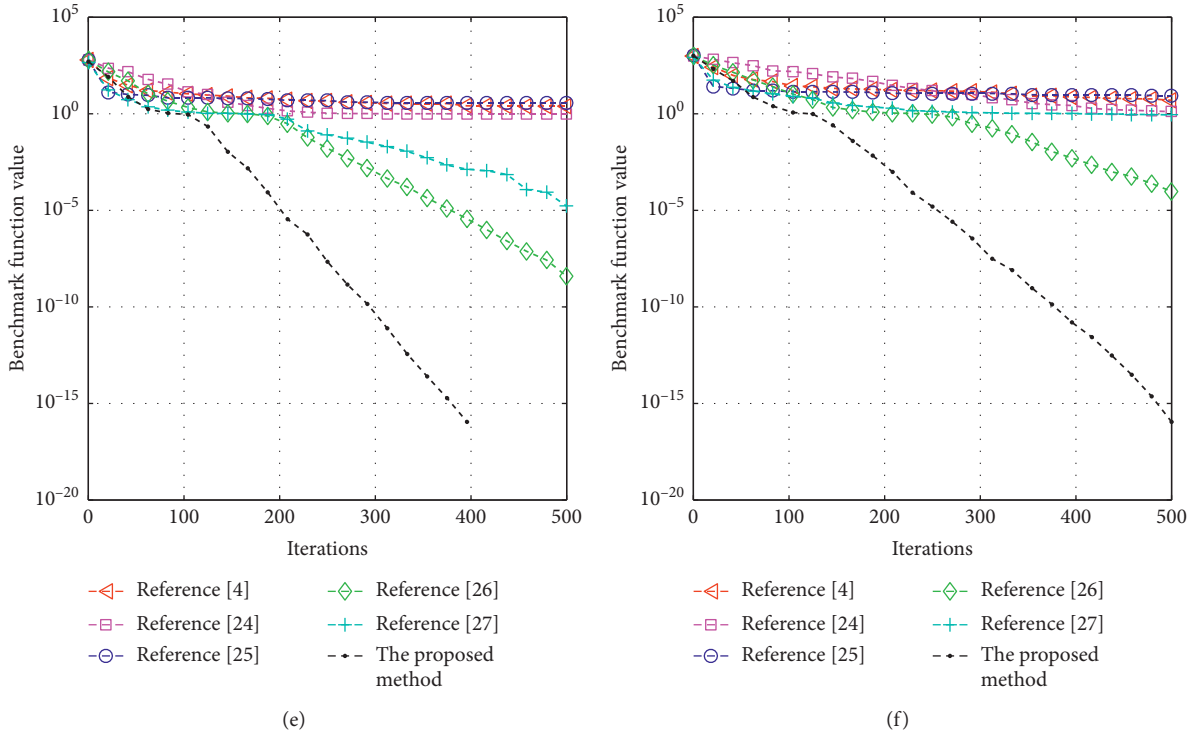
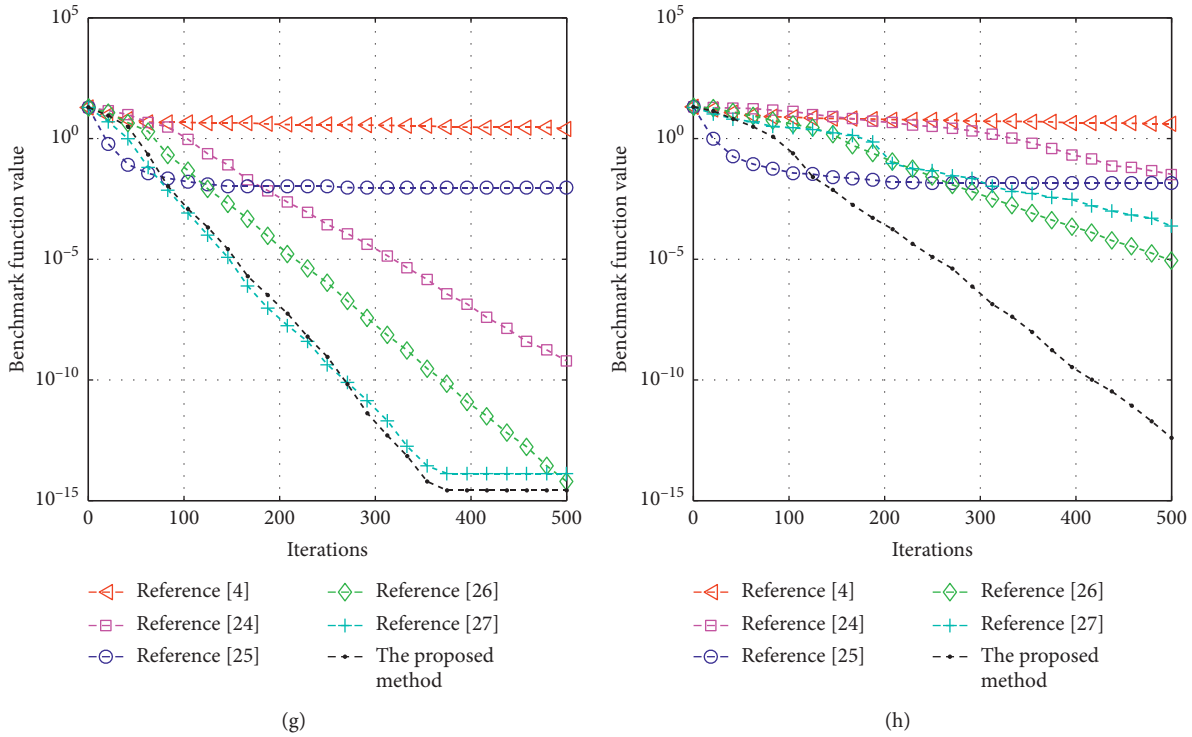


Figure 3: Continued.



(e) (f)



(g) (h)

Figure 3: Continued.

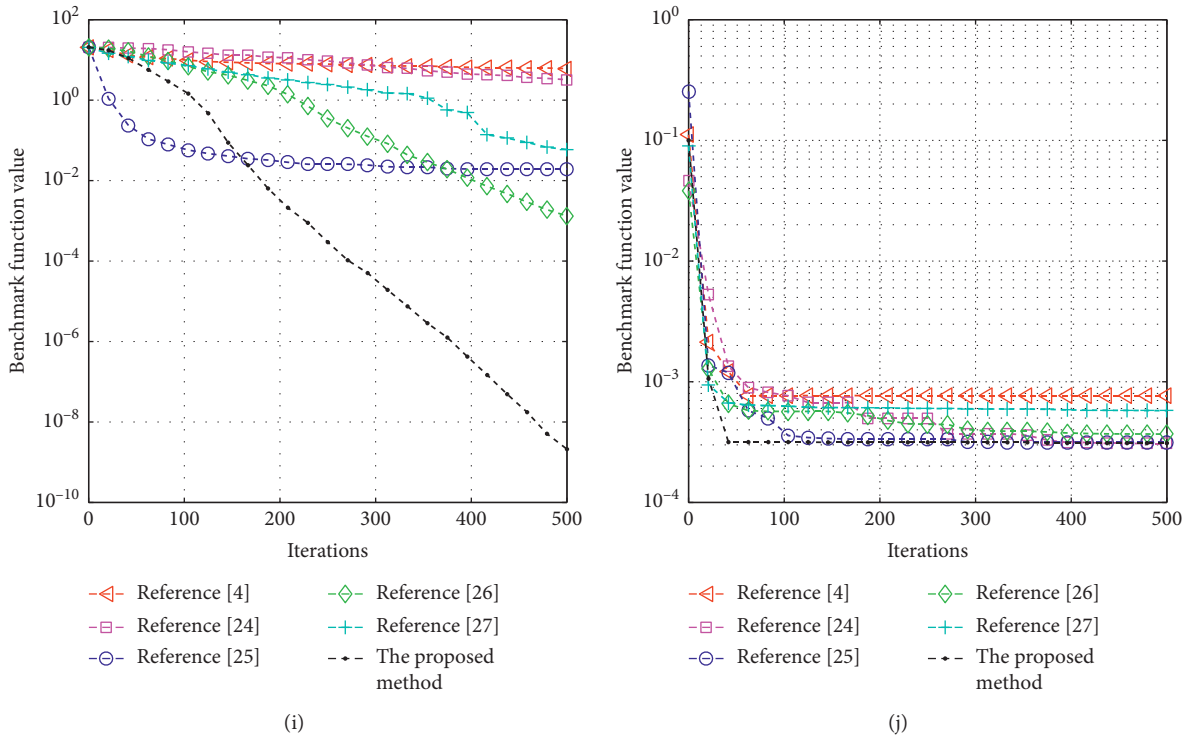


Figure 3: Multimodal function simulation results diagram, (a) f_5: Alpine$(d = 10)$, (b) f_5: Alpine$(d = 30)$, (c) f_5: Alpine$(d = 50)$, (d) f_6: Griewank$(d = 10)$, (e) f_6: Griewank$(d = 30)$, (f) f_6: Griewank$(d = 50)$, (g) f_7: Ackley$(d = 10)$, (h) f_7: Ackley$(d = 30)$, (i) f_7: Ackley$(d = 50)$, (j) f_8: Kowalik$(d = 4)$.

Table 5: Wilcoxon rank sum test results of unimodal function.

Function	Dimension	NBBO vs BBO		NBBO vs DE		NBBO vs EHO		NBBO vs DEBBO		NBBO vs BBOPSO	
		P	W	P	W	P	W	P	W	P	W
$f_1(x)$ Quartic	$D = 10$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$1.40e-03$	+	$1.68e-06$	+
	$D = 30$	$2.82e-11$	+	$2.82e-11$	+	$2.82e-11$	+	$2.82e-11$	+	$6.93e-11$	+
	$D = 50$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.82e-11$	+	$2.87e-11$	+
$f_2(x)$ Sphere	$D = 10$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.78e-11$	+
	$D = 30$	$2.87e-11$	+	$2.87e-11$	+	$2.82e-11$	+	$2.87e-11$	+	$2.87e-11$	+
	$D = 50$	$2.87e-11$	+	$2.87e-11$	+	$2.82e-11$	+	$2.87e-11$	+	$2.87e-11$	+
$f_3(x)$ Powell	$D = 10$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
	$D = 30$	$2.87e-11$	+	$2.87e-11$	+	$2.93e-11$	+	$2.93e-11$	+	$2.93e-11$	+
	$D = 50$	$2.87e-11$	+	$2.87e-11$	+	$2.78e-11$	+	$2.87e-11$	+	$2.87e-11$	+
$f_4(x)$ Easom	$D = 2$	1.17e−12	+	N/A	N/A	1.17e−12	+	N/A	N/A	N/A	N/A

Table 6: Wilcoxon rank sum test results of multimodal function.

Function	Dimension	NBBO Vs BBO		NBBO vs DE		NBBO vs EHO		NBBO vs DEBBO		NBBO vs BBOPSO	
		P	W	P	W	P	W	P	W	P	W
$f_5(x)$ Alpine	$D = 10$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
	$D = 30$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
	$D = 50$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
$f_6(x)$ Griewank	$D = 10$	$1.17e-12$	+	$1.69e-14$	+	$1.17e-12$	+	$5.62e-11$	+	$5.62e-11$	+
	$D = 30$	$1.17e-12$	+	$1.69e-14$	+	$1.17e-12$	+	$1.17e-12$	+	$1.17e-12$	+
	$D = 50$	$3.06e-12$	+	$3.06e-12$	+	$3.06e-12$	+	$3.06e-12$	+	$3.06e-12$	+
$f_7(x)$ Ackley	$D = 10$	$1.17e-12$	+	$1.17e-12$		$1.03e-13$	+	N/A	N/A	$1.15e-12$	+
	$D = 30$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
	$D = 50$	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+	$2.87e-11$	+
$f_8(x)$ Kowalik	$D = 2$	$2.87e-11$	+	$4.27e-07$	+	$1.68e-06$	+	$2.31e-08$	+	$2.87e-11$	+

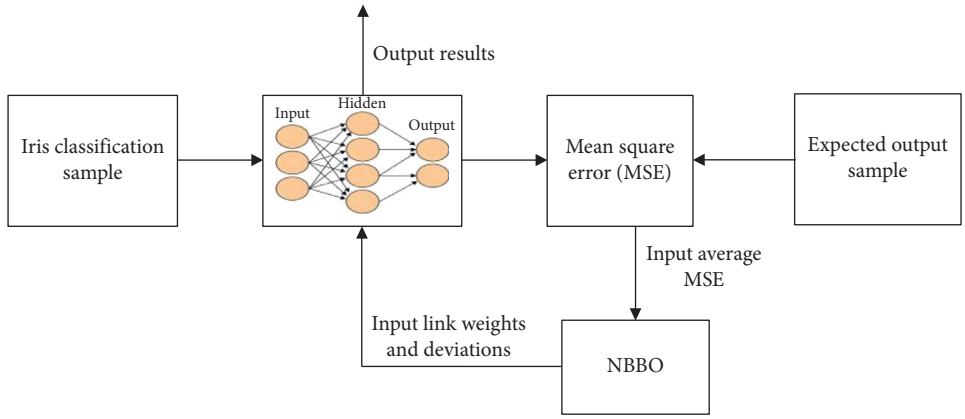


Figure 4: NBBO-based Iris classification diagram.

Table 7: Simulation results of iris classification.

Algorithm	Optimal value	Average value	Standard deviation
Reference [4]	3.15 $E-02$	5.34 $E-02$	1.52 $E-02$
Reference [24]	6.12 $E-02$	1.31 $E-01$	3.48 $E-02$
Reference [25]	2.37 $E-01$	3.92 $E-01$	7.36 $E-02$
Reference [26]	2.73 $E-02$	4.72 $E-02$	1.29 $E-02$
Reference [27]	1.51 $E-01$	2.56 $E-01$	5.34 $E-02$
The proposed method	1.34 E − 02	1.93 E − 02	3.80 E − 03

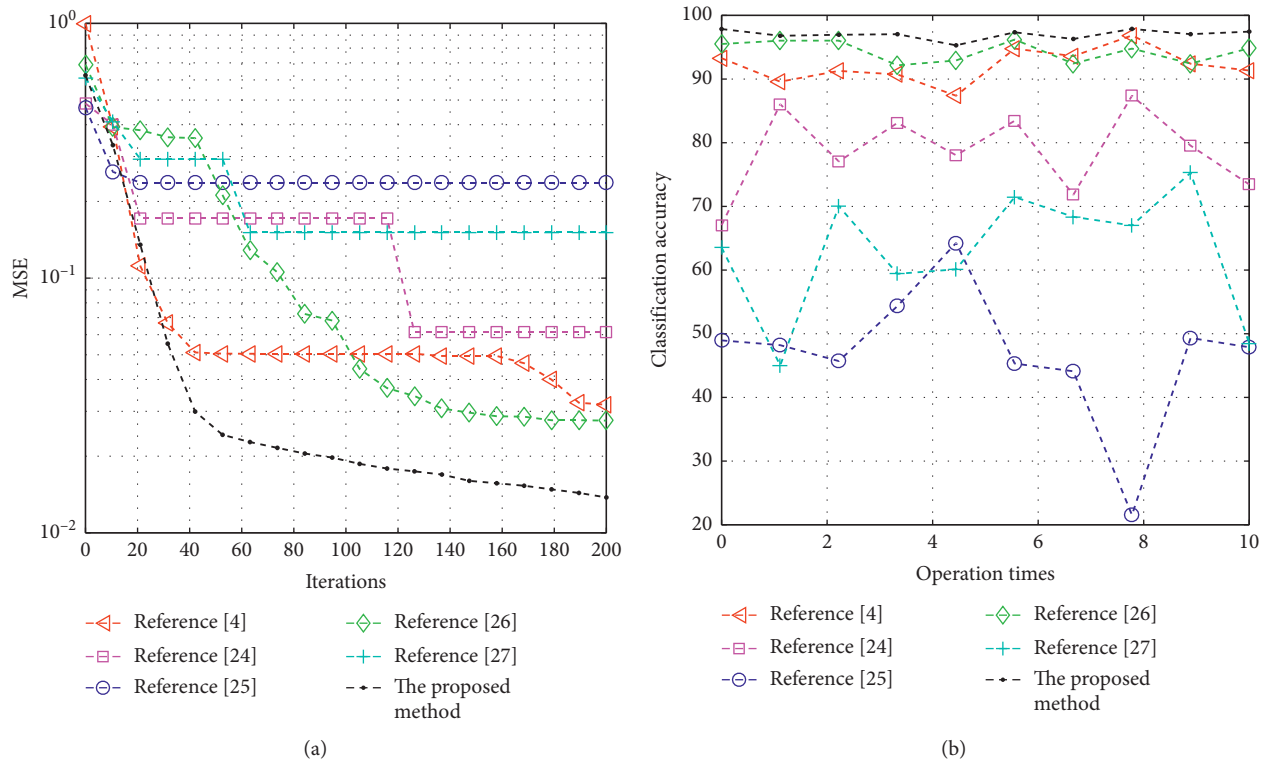


Figure 5: (a) Convergence curve and (b) classification accuracy.

through continuous iterative evolution, so as to provide the optimal connection weight and deviation for iris classification samples after iterative training.

The iris dataset has 150 training samples divided into 3 categories: Setosa, Versicolor, and Virginica, with a total of 4 basic features: sepal length, sepal width, petal length, and petal

width. We used the structure of multilayer perceptron as 4-9-3 to classify the problem. Table 7 shows the experimental results of the data set algorithm. Figure 5 shows the average convergence trend and classification accuracy.

Compared with other five iris classification algorithms in Table 7, we can see that the proposed NBBO has the better optimal value, average value, and standard deviation than the other five algorithms. The optimal and average values of the MSE calculated by NBBO are better than other algorithms, which indicates that the optimization performance of the NBBO is the best among the seven algorithms. At the same time, the minimum standard deviation calculated by NBBO also indicates that the NBBO algorithm inherits the strong robustness of the BBO algorithm and is more robust than other algorithms.

Figure 5(a) shows the classification convergence curves of the six algorithms for the iris classification problem, among which the convergence effect of the DE algorithm is the worst of the six algorithms, and the EHO and BBOPSO algorithms fall into the local optimum in the late iteration. BBO algorithm and DEBBO algorithm have similar convergence precision but the convergence speed is not ideal. The proposed NBBO algorithm is optimal in convergence precision and convergence speed. Figure 5(b) shows that DE, EHO, and BBOPSO show significant fluctuations in the classification accuracy of 10 independent operations. The NBBO algorithm has the best stability and average, and its classification accuracy is the highest among the three stable algorithms BBO, DEBBO, and NBBO.

4. Conclusions and Future Work

This paper proposed a novel biogeography-based optimization algorithm based on local migration strategy and nonuniform mutation for iris classification. Firstly, the hyperbolic cotangent transfer model was used to replace the original transfer model to enhance the adaptability of the algorithm. Secondly, the local transfer strategy was used to improve the global search ability of the algorithm. Finally, the nonuniform mutation operation was used to enhance the development ability of the algorithm in the later stage of iteration for the optimal solution in a small range, to improve the ability of the algorithm to exit the local optimum. The simulation results showed that the proposed NBBO has higher stability, better convergence accuracy, and faster convergence speed than other method for iris classification. How to use NBBO algorithm to solve other multiobjective optimization problem compare with multiobjective optimization method [30–33] and realize better applications for engineering practice will be the focus of our next work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Research Programme of Colleges and Universities of Anhui Province under grant KJ2020ZD39, the Foundation for Talented Young People of Anhui Polytechnic University under grant 2016BJRC008, the Open Research Fund of Anhui Key Laboratory of Detection Technology and Energy Saving Devices, and Anhui Polytechnic University under grant DTESD2020A02.

References

[1] S. A. C. Schuchkers, N. A. Schmid, A. Abhyankar et al., "On techniques for angle compensation in nonideal iris recognition," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1083–4419, 2017.

[2] S. Gao, X. D. Zhu, and Y. N. Liu, "A quality assessment method of iris image based on support vector machine," *Journal of Fiber Bioengineering & Informatics*, vol. 8, no. 2, pp. 293–330, 2015.

[3] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *Pattern Recognition*, vol. 36, no. 8, pp. 1783–1797, 2003.

[4] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[5] X. Zhang, D. Wang, and H. Chen, "Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation," *IEEE Access*, vol. 7, pp. 28810–28825, 2019.

[6] F. Zhao, S. Qin, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A hybrid biogeography-based optimization with variable neighborhood search mechanism for No-wait flow shop scheduling problem," *Expert Systems with Applications*, vol. 126, no. 6, pp. 321–339, 2019.

[7] M. Kaveh and M. S. Mesgari, "Improved biogeography-based optimization using migration process adjustment: an approach for location-allocation of ambulances," *Computers & Industrial Engineering*, vol. 135, pp. 800–813, 2019.

[8] B. X. Li and K. S. Low, "Low sampling rate online parameters monitoring of DC-DC converters for predictive-maintenance using biogeography-based optimization," *IEEE Transactions on Power Electronics*, vol. 31, no. 4, pp. 2870–2879, 2016.

[9] Y. B. Bhushan, S. U. Bhimrao, and R. D. Koteswara, "An efficient transient stability-constrained optimal power flow using biogeography-based algorithm," *International Transactions on Electrical Energy Systems*, vol. 28, no. 1, pp. 1–15, 2018.

[10] A. Kaveh, A. Dadras, and N. G. Malek, "Buckling load of laminated composite plates using three variants of the biogeography-based optimization algorithm," *Acta Mechanica*, vol. 229, no. 4, pp. 1551–1566, 2018.

[11] J. S. Wang and J. D. Song, "Chaotic biogeography-based optimization algorithm," *IAENG International Journal of Computer Ence*, vol. 22, no. 2, pp. 122–134, 2017.

[12] Q. X. Feng, S. Y. Liu, J. K. Zhang et al., "Improved biogeography-based optimization with random ring topology and Powell's method," *Applied Mathematical Modelling*, vol. 41, no. 6, pp. 630–649, 2017.

[13] H. Li and Y. H. Liu, "Beam pattern synthesis based on improved biogeography-based optimization for reducing

sidelobe level," *Computers and Electrical Engineering*, vol. 60, no. 5, pp. 161–174, 2017.

[14] X. Chen and K. J. Yu, "Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters," *Solar Energy*, vol. 180, pp. 192–206, 2019.

[15] F. Zhao, S. Qin, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A two-stage differential biogeography-based optimization algorithm and its performance analysis," *Expert Systems with Applications*, vol. 115, pp. 329–345, Jan, 2019.

[16] W. A. Guo, M. Chen, L. Wang et al., "A survey of biogeography-based optimization," *Neural Computing & Applications*, vol. 28, no. 8, pp. 1909–1926, 2017.

[17] A. R. AI-Roomi and M. E. EI-Hawary, "Metropolis biogeography-based optimization," *Information Sciences*, vol. 360, no. 5, pp. 73–95, 2016.

[18] H. P. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.

[19] S. A. Darani and O. Abdelkhalik, "Convergence analysis of hidden genes genetic algorithms in space trajectory optimization," *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 228–238, 2018.

[20] B. X. Yue, H. B. Liu, and A. Abraham, "Dynamic trajectory and convergence analysis of swarm algorithm," *Computing & Informatics*, vol. 31, no. 2, pp. 371–392, 2012.

[21] F. A. Attia, "Resolvent operators of Markov processes and their applications in the control of a finite dam," *Journal of Applied Probability*, vol. 26, no. 2, pp. 314–324, 1989.

[22] D. Ebenezer, "Optimum wavelet-based homomorphic medical image fusion using hybrid genetic–grey wolf optimization algorithm," *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6804–6811, 2018.

[23] Q.-K. Pan, H.-Y. Sang, J.-H. Duan, and L. Gao, "An improved fruit fly optimization algorithm for continuous function optimization problems," *Knowledge-Based Systems*, vol. 62, pp. 69–83, 2014.

[24] R. Storn and K. Price, ""Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[25] G. G. Wang, L. Dos Santos Coelho, X. Z. Gao, and S. Deb, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour," *International Journal of Bio-Inspired Computation*, vol. 8, no. 6, pp. 394–409, 2016.

[26] W. Y. Gong, Z. H. Cai, and C. X. Ling, "DE/BBO: A hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.

[27] D. Li, W. Guo, L. Wang et al., "Particle swarm optimization-based solution updating strategy for biogeography-based optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016.

[28] D. Joaquín, G. Salvador, D. Molina et al., "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm & Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[29] H. Mannila, "Data mining: machine learning, statistics, and databases," in *Proceedings of 8th International Conference on Scientific and Statistical Data Base Management*, June 1999.

[30] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[31] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[32] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[33] K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," *IEEE Transactions on Cybernetics*, p. 1. In press, 2020.

Hindawi
Complexity
Volume 2021, Article ID 8843271, 12 pages
https://doi.org/10.1155/2021/8843271

Research Article

Topology-Aware Bus Routing in Complex Networks of Very-Large-Scale Integration with Nonuniform Track Configurations and Obstacles

Ziran Zhu,[1] Zhipeng Huang,[2] Jianli Chen,[3] and Longkun Guo [4]

[1]*National ASIC System Engineering Research Center, Southeast University, Nanjing, China*
[2]*Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China*
[3]*State Key Laboratory of ASIC and System, Fudan University, Shanghai, China*
[4]*Shandong Key Laboratory of Computer Networks, School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China*

Correspondence should be addressed to Longkun Guo; longkun.guo@gmail.com

Received 9 September 2020; Revised 21 January 2021; Accepted 26 March 2021; Published 15 April 2021

Academic Editor: Shi Cheng

As one of the most important routing problems in the complex network within a very-large-scale integration (VLSI) circuit, bus routing has become much more challenging when witnessing the advanced technology node enters the deep nanometer era because all bus bits need to be routed with the same routing topology in the context. In particular, the nonuniform routing track configuration and obstacles bring the largest difficulty for maintaining the same topology for all bus bits. In this paper, we first present a track handling technique to unify the nonuniform routing track configuration with obstacles. Then, we formulate the topology-aware single bus routing as an unsplittable flow problem (UFP), which is integrated into a negotiation-based global routing to determine the desired routing regions for each bus. A topology-aware track assignment is also presented to allocate the tracks to each segment of buses under the guidance of the global routing result. Finally, a detailed routing scheme is proposed to connect the segments of each bus. We evaluate our routing result with the benchmark suite of the 2018 CAD Contest. Compared with the top-3 state-of-the-art methods, experimental results show that our proposed algorithm achieves the best overall score regarding specified time limitations.

1. Introduction

As the advanced technology node enters the deep nanometer era, routing has become much challenging because of the enormously growing scale of the large scale of very-large-scale integration (VLSI) circuit [1]. Among the routing problems, bus routing is attracting most research interest and has met new challenges: (1) all bits in each bus must be routed with the same routing topology; (2) nonuniform and complex routing track configurations; and (3) we need to handle obstacles. Particularly, the constraint according to which all bits belonging to the same bus must be routed following the same routing topology makes the previous routers not applicable to current topology-matching bus routing.

Previous works on bus routing focused mainly on printed circuit board (PCB) designs. For example, Tian and Watanabe [2] considered the delay-matching constraint in bus routing to meet several timing specifications. Yan and Wong [3] and Zhang et al. [4] handled the length-matching bus routing such that the wire lengths of all nets on the same bus are within the specified range. However, none of these works considered the constraint of maintaining the same topology for all bits on the same bus. Therefore, it is desirable to develop an effective and efficient topology-matching bus routing algorithm.

According to the previous criteria, the bits in a bus are considered to have the same topology if the following four criteria are met: (1) All bits have the same number of wires. (2) All wires traced from all bits have the same layer

sequencing. (3) All wires traced from all bits route towards the same direction. (4) Within each segment (a segment is a set of wires of different bits with the same sequence when traced from a set of pin shapes), the wires of different bits maintain the same or the reverse order as the order seen from the pin shapes.

Figure 1(a) illustrates a bus being routed successfully with the same topology. Supposing that we start tracking the wires from the pin shapes on the left, as shown in this figure, all wires traced from all bits have the same direction (rightward, downward, and rightward, resp.) and the same layer sequencing (L1, L2, and L1). In addition, within all segments (seg1, seg2, and seg3), the wires of different bits maintain the same order or the reverse order as the order of pin shapes.

Routing tracks are designed to help routers comply with various design requirements and help mask coloring, which are essential in advanced technology node [5]. Each routing track has a width constraint that only wires with width no larger than the constraint are allowed to be routed on the track. Since the routing requirements of different buses may be different (e.g., different wire widths and different wire spacing), the routing track configuration may be nonuniform. For example, Figure 1(b) shows five tracks that can be classified into two types (blue and green, resp.) based on the width constraints. The blue tracks have a larger width constraint and cover the whole design from bottom to top, while the green tracks have a smaller width constraint and only cover part of the design. Particularly, routing tracks can overlap with each other, and the distribution may be uneven. Such nonuniform routing track configuration imposes a great challenge to the bus routing.

Obstacles such as circuit components and power vias make bus routing even more challenging. Since such obstacles are scattered throughout certain layers, it is not possible to find continuous routable area if bus bits are not allowed to route between some of them.

Due to the high complexity of the routing problems, the routing process is typically divided into global routing, track assignment, and detailed routing. In global routing, the routing region is divided into coarse-grained grid cells (called g-cells), and rough routing regions are determined for each net through the connection between the g-cells. Next, track assignment allocates routing tracks to iroutes that are extracted from the global routing result. Finally, detailed routing finds a path for each net to connect the iroute and pins and completes the final routing.

In this paper, we propose an effective algorithm to solve the topology-matching bus routing problem considering obstacles and nonuniform track configurations. The major contributions of our work are summarized as follows:

(i) A track handling technique is presented to unify the nonuniform routing track configuration with obstacles.

(ii) We formulate the topology-aware single bus routing as an unsplittable flow problem (UFP), which is integrated into a negotiation-based global routing to determine the desired routing regions for each bus.

(iii) Under the guidance of the global routing result, a topology-aware track assignment is proposed to allocate tracks to each segment of buses, which significantly reduces the difficulty of maintaining the same routing topology in the subsequent steps.

(iv) A detailed routing scheme considering routing topology for all bus bits is presented to connect the segments of each bus.

(v) Experimental results show that our proposed bus routing algorithm is effective. Compared with the top 3 teams of the CAD Contest at ICCAD on Obstacle-Aware On-Track Bus Routing [5], our proposed algorithm can achieve the best overall score within the specified time.

The rest of this paper is organized as follows. Section 2 first introduces the bus routing preference metrics and then gives the problem statement. Section 3 details our bus routing algorithm. Section 4 provides the experimental results. Finally, conclusions are drawn in Section 5.

2. Preliminaries

In this section, we first introduce the bus routing preference metrics considered in the 2018 CAD Contest at ICCAD [5] and then formulate the topology-matching bus routing problem.

2.1. Bus Routing Preference Metrics. For successfully routed buses, the metrics such as wire length, the number of segments, the width of each segment, and the number of spacing violations are used to measure the bus routing quality in the contest [5]. We detail these four metrics as follows.

2.1.1. Wire Length. Wire length is a basic metric of routing quality. Longer wire lengths typically imply larger delays and larger power consumption [6]; hence routers are expected to minimize the wire length. The wire length of a bus is calculated by summing up the wire length of all bits, and taking detour will cause an increase in wire length.

2.1.2. Number of Segments. Since each layer has a preferred routing direction (either horizontal or vertical), more segments indicate that more vias are used. However, vias are undesirable due to their negative impacts on signal integrity, delay, routing area, and manufacturing yields [6]. Therefore, an ideal bus router should minimize the number of segments.

2.1.3. Width of Segments. If the direction of a segment is horizontal, then the width of the segment is defined as the y-coordinate of the topmost wire in the segment minus the y-coordinate of the bottommost wire in the segment; in contrast, if the direction of a segment is vertical, the

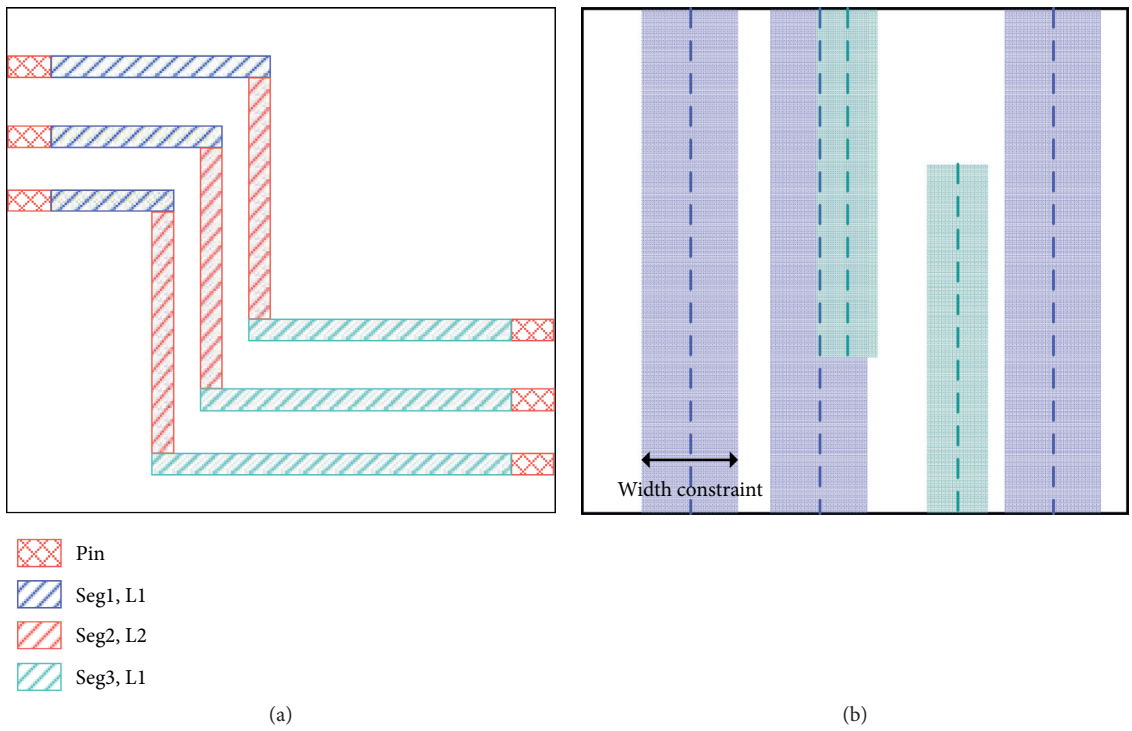


Figure 1: (a) A bus with the same routing topology. (b) Nonuniform routing track configuration.

calculation is between the x-coordinate of the rightmost and the leftmost wire [5]. The smaller the width of a segment is, the more compact the corresponding bus is.

2.1.4. Spacing Violation. Each layer has a spacing constraint which specifies the minimum distance that should be maintained between the routing paths of a bit and the design boundary, the obstacles, and other routing wires on that layer. Spacing violation will result in an additional penalty in the evaluation score.

Figure 2 shows an example of bus routing preference metrics, where Figure 2(a) gives an inferior routing result with a longer wire length, a large number of segments, and a larger width of segments, while Figure 2(b) gives a desired solution because its wire length and number of segments are minimized, and the width of segments is also smaller.

2.2. Problem Statement. We are given the following: (1) A design with l routing layers $\mathscr{L} = \{L_1, L_2, \ldots, L_l\}$. Each layer has a routing direction and a spacing constraint. The routing direction is either vertical or horizontal, and the spacing constraint specifies the minimum distance that should be maintained between the output routing path and the design boundary, the obstacles, and other routing wires in that layer. (2) A set of m routing tracks $\mathscr{T} = \{T_1, T_2, \ldots, T_m\}$. Each track T_i is represented by a line in a layer with a width constraint WT_i. The direction of each track is always the same as the routing direction of the layer that the track is on, and the width constraint requires that only the wires with a width smaller than or equal to the width constraint can be routed on the track. (3) A set of p obstacles $\mathscr{O} = \{O_1, O_2, \ldots, O_p\}$, which are scattered throughout certain layers. (4) A set of n buses $\mathscr{B} = \{B_1, B_2, \ldots, B_n\}$. Each bus B_i consists of NB_i bits, all bits have NP_i pin shapes, and the width constraint WB_{ij} $(1 \le j \le l)$ indicates that the routing paths for the bus B_i in layer L_j have the width equal to the width constraint.

The goal of topology-matching bus routing is to achieve as many successfully routed buses as possible, and the following metrics should also be minimized simultaneously: (1) the total wire length of all buses; (2) the number of segments; (3) compactness of each bus (i.e., the width of segments); (4) the number of spacing violations.

A bus is routed successfully if the following three hard constraints are met: (1) Routing paths of each bit connect all pin shapes of the bit and do not overlap with the paths of other bits. (2) All wires are on-track without violating the width constraint of the tracks and do not overlap with obstacles. (3) All bits are routed with the same topology.

3. Our Algorithm

The overall flow of our proposed algorithm is summarized in Figure 3, which mainly consists of four parts: (1) preprocessing, (2) global routing, (3) track assignment, and (4) detailed routing.

The preprocessing stage unifies the nonuniform routing track configuration with obstacles to support efficient query and simplify subsequent routing operations. In the global routing stage, we formulate the topology-aware single bus routing as UFP and integrate it into a negotiation-based global routing to determine the desired routing regions for each bus. The complexity of subsequent steps can be reduced

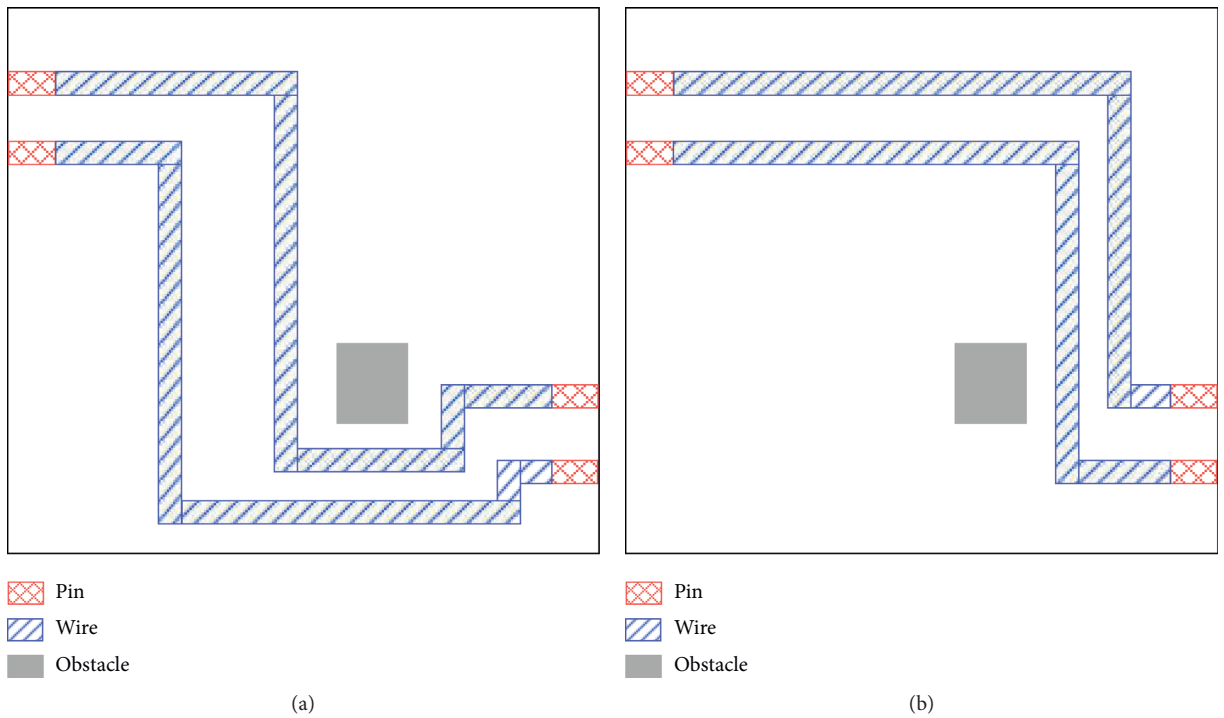


Figure 2: Example of bus routing preference metrics. (a) An inferior routing result. (b) A desired routing result.

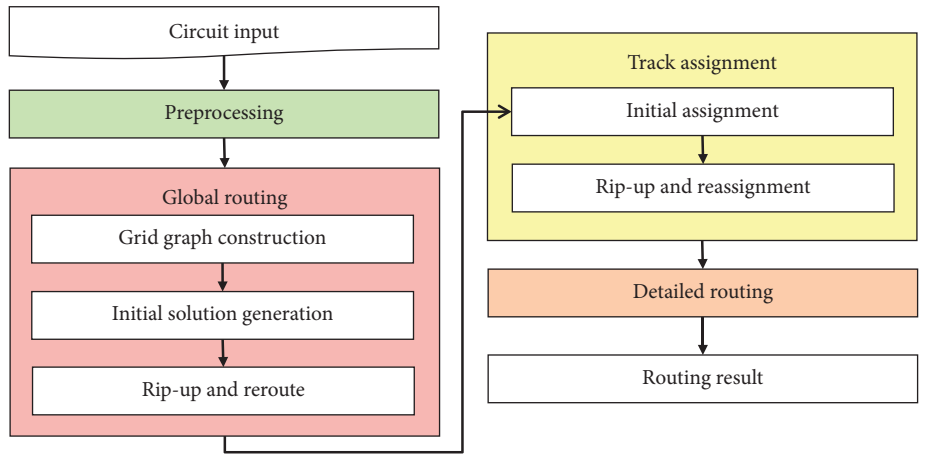


Figure 3: Our algorithm flow.

by confining its search space to the regions identified by the global routing stage. Under the guidance of the global routing result, the track assignment stage allocates tracks to each segment of buses, which significantly reduces the difficulty of maintaining the same routing topology in the subsequent step. Finally, the detailed routing stage connects the segments of each bus bit and obtains the final routing result. We shall detail these four major parts in the following subsections.

3.1. Preprocessing. In this subsection, we perform some preprocessing on the input data to simplify subsequent routing operations. Since the nonuniform routing track configuration and obstacles make the bus routing much more complicated, we first present a track handling technique to unify the track configuration while considering obstacles. Specifically, we treat each routing track uniformly as covering the entire design from top to bottom (or left to right), and each track has a set of intervals for recording the subtracks that have been used. Furthermore, if the centerlines of two tracks overlap, we will shrink or delete the used intervals of the track that has a smaller width.

Figure 4 shows three nonuniform routing tracks and an obstacle. In the figure, we treat these three tracks as covering the entire design from top to bottom, and tracks T_1, T_2, and T_3 have the sets of used intervals $\{I_1, I_2\}$, $\{I_4\}$, and $\{I_3\}$,

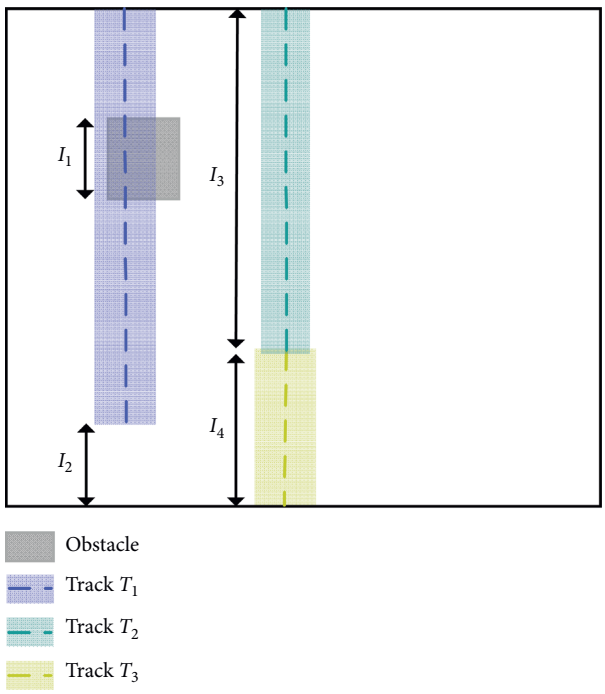


Figure 4: Example of our track handling technique.

respectively. The interval I_1 is occupied by the obstacle, and intervals I_2, I_3, I_4 are not covered by the corresponding tracks. In addition, since the centerlines of tracks T_2 and T_3 overlap and the track T_2 has a smaller width, we update the set of used intervals of track T_2 by deleting the interval I_4. Then the set of used intervals of track T_2 is finally empty, and thus the wires can go through directly from track T_2 to track T_3.

Besides, we adopt a minimum spanning tree algorithm to decompose each multipin bit into a set of two-pin bits and determine a preferred direction (horizontal or vertical) for each pin. That is, if the physical locations of the same pin shapes in different bits are horizontally distributed, then the preferred directions of the pin shapes are set as vertical; in contrast, if the physical positions of the same pin shapes in different bits are vertically distributed, then the preferred directions of the pin shapes are horizontal. The preferred direction of a pin is the desired direction of the wire that connects to the pin. For example, the preferred directions of all six pins in Figure 1(a) are horizontal, since the physical positions of the same pin shapes in different bits are vertically distributed.

3.2. Global Routing. To determine the desired routing regions for each bus and reduce the complexity of subsequent detailed routing, we formulate the topology-aware single bus routing as UFP and integrate it into a negotiation-based global routing scheme. The three main steps of our global routing scheme are elaborated as follows.

3.2.1. Grid Graph Construction. In the global routing stage, each routing layer is partitioned into a set of global cells (g-cells) as shown in Figure 5(a), and a corresponding grid graph can be constructed as shown in Figure 5(b). In the grid graph, each vertex represents a g-cell and each routing edge represents a boundary between adjacent g-cells, and any two adjacent layers are connected by vias. In addition, the number on each routing edge of Figure 5(b) indicates the capacity of the edge, which corresponds to the number of routing tracks that can be contained across the edge. Since solving the 3D global routing problem directly is time-consuming, we further project a multilayered design onto the 2D plane, and then a capacitated graph $G(V, E, u)$ is constructed.

Besides, each pin corresponds to a g-cell. If two pins of any two bits are in the same g-cells, then we temporarily combine the two bits as a bit. In this way, we can reduce the number of bits in each bus greatly in global routing, and the runtime of the global routing stage will be reduced. Take Figure 5(a) for example. There are two bits in a bus and each bit has two pins. Since the two pins of two bits are in the same g-cells, we combine the two bits as a bit, and the demand (number of tracks consumed) of each routed wire in the merged bit is 2.

3.2.2. Initial Solution Generation. Too many bends of a routing path not only increase the number of vias, resulting in poor routing quality, but also make it more difficult to maintain the same routing topology due to the increase of the number of segments. Therefore, we limit the number of bends in this initial solution generation step. The bits can be categorized into two types based on the preferred directions of pins. Figure 6(a) shows the pins of four bits from different buses with the same preferred direction, and the path connecting each bit has an even number of bends. Conversely, the pins of three bits from different buses shown in Figure 6(b) have orthogonally preferred directions, and the path connecting the two pins of each bit has an odd number of bends. Further, since we set a preferred direction for each pin, a bit has at most one path with one or zero bends, and we only need to determine $n-1$ $(n \geq 2)$ bending points in turn to obtain a path with n bends.

In this step, the number of bends of a path connecting each bit is limited to four. For each bus, let d_i represent the demand of bit i, $\mathscr{P}_i$ denote the set of paths for bit i, and $\mathscr{P}_i^T$ represent the set of paths with the same routing topology T in $\mathscr{P}_i$. Note that, the topology in global routing ensures that (1) all bits have the same number of wires and (2) all wires traced from all bits route towards the same direction, while temporarily ignoring the relative order of the wire of different bits. In addition, for each $P \in \mathscr{P}_i$, we have a non-negative variable $x(P)$ and a weight $w(P)$ associated with it. The weight $w(P)$ of path P is the sum of the weights of all the edges on the path, and the weight of edge e is defined as

$$w(e) = \frac{1}{1 + e^{(d(e)-u(e))}}, \tag{1}$$

where $d(e)$ represents the sum of the demands of the bits passing through e and $u(e)$ is the capacity of the edge e. For each edge e, the demand $d(e)$ is initialized to 0 and is updated once a bus is routed successfully. The weight $w(e)$

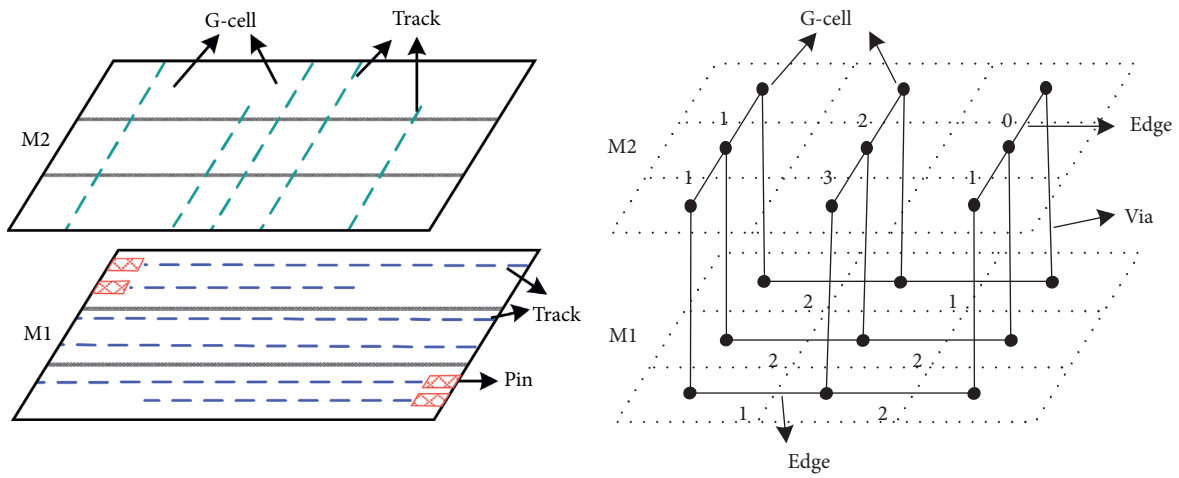


Figure 5: Example of grid graph construction. (a) Each routing layer is partitioned into a set of global cells (g-cells). (b) Each g-cell is modeled as a vertex, and abutting g-cells are connected by routing edges.

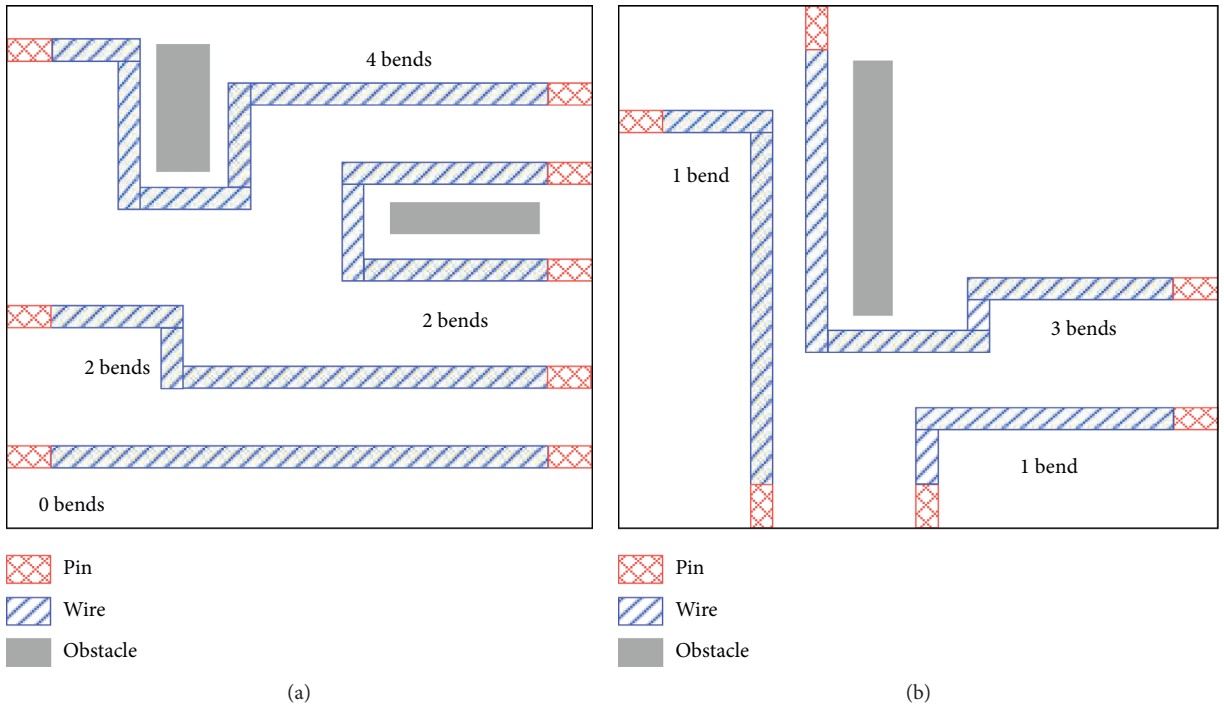


Figure 6: The bits can be categorized into two types based on the preferred directions of pins. (a) The two pins of each bit have the same preferred direction, and the path connecting the two pins has an even number of bends. (b) The two pins of each bit have orthogonally preferred directions, and the path connecting the two pins has an odd number of bends.

decreases dramatically as the demand approaches the capacity but grows slowly in the undercapacity and overcapacity parts.

We determine the order of routing topology according to the number of bends and the weight of path $w(P)$. A smaller number of bends have a higher priority. For each bus, we try the topologies of the bus one by one until the bus is routed successfully. Further, we introduce a new variable x_i for each bit i, where $x_i = \sum_{P \in \mathcal{P}_i^T} x(P)$, and let u' be a copy of u. Then, the global routing problem of the bus with topology T can be formulated as UFP as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n_{\text{bit}}} \sum_{P \in \mathcal{P}_i^T} w(P) \cdot x(P) \\
& x_i - \sum_{P \in \mathcal{P}_i^T} x(P) = 0, \quad 1 \le i \le n_{\text{bit}} \\
\text{s.t.} \quad & \sum_{i=1}^{n_{\text{bit}}} d_i \sum_{P \in \mathcal{P}_i^T:\, e \in P} x(P) \le u'(e) - d(e), \quad e \in E \\
& x_i, x(P) \in \{0,1\}, \quad 1 \le i \le n_{\text{bit}}, P \in \cup_i \mathcal{P}_i^T.
\end{aligned} \tag{2}$$

In the formulated UFP, the objective is to maximize the number of routable bits, and the total weight of all selected paths is as large as possible (i.e., the congestion is as small as possible). The constraints in the first line ensure that at most one path is selected per bit, and the constraints of the second line limit the total demand of bits that can pass through each edge. u' will be increased if all the topologies of the bus fail to be routed. A bus is successfully routed if all bits of the bus are successfully routed (i.e., $x_i = 1, 1 \le i \le n_{\text{bit}}$). Once a bus is successfully routed, we update the demand $d(e)$ and weight $w(e)$ of each edge e and then handle the next bus.

Based on the combinatorial algorithm for UFP in [7], Algorithm 1 provides an algorithm for problem (1). Let $|E| = m$ and $u_{\min}(u_{\max})$ be the minimum (maximum) edge capacity and $d_{\min}, d_{\max}, w_{\min}, w_{\max}$ be the minimum/maximum demand/weight among all bus bits. In Line 1, we first partition the set of bits T into two disjoint sets T_1 and T_2. T_1 consists of bits for which $d_j \le u_{\min}/2$, and the rest of the bits are in T_2. For each bit j and a given path P of bit j, we adopt $F(j,P)$ in [7] to measure the weight gain relative to the added demand load. We set the lower bound α_{lb} and the upper bound α_{ub} on F in Line 3. The order of bits are sorted in Line 6, and then we handle the bits one by one to select a path for each bit in Lines 7–11. $L_{j-1}(e)$ in Line 8 denotes the relative load of edge e after routing bit j.

The time complexity of Algorithm 1 is $O(n_{\text{bit}} \cdot |E_{P^T}|)$, where n_{bit} is the number of bits in a bus and $|E_{P^T}|$ is the number of edges of the paths that have the same routing topology T for the bus. In detail, as can be seen from Algorithm 1, Line 1 requires $O(n_{\text{bit}})$ time, Line 6 requires $O(n_{\text{bit}} \cdot \log(n_{\text{bit}}))$, and Lines 8–9 need $O(|E_{P^T}|)$ time. Besides, since the number of loops in Line 2 and Line 4 is constants, the number of loops in Line 7 is n_{bit}. Hence, Algorithm 1 requires $O(n_{\text{bit}} \cdot |E_{P^T}|)$ time for each bus.

Theorem 1. *Algorithm 1 is an $O(\sqrt{m})$ approximation algorithm for the UFP.*

Proof. Consider an optimal solution routing bits in $\mathcal{Q} \subseteq T$. For each $j \in \mathcal{Q}$, let $\mathcal{Q}_j$ be the route chosen for j in the optimal solution. The total weight of either $\mathcal{Q} \cap T_1$ or $\mathcal{Q} \cap T_2$ is at least $w(\mathcal{Q}/2)$. Denote that set by $\mathcal{Q}'$ and its index by $i' \in \{1,2\}$, and let $\alpha' = 2^{k'}$ be the highest such that $w(\{j \in \mathcal{Q}' | F(j, \mathcal{Q}_j) > \alpha'\}) \ge w(\mathcal{Q})/4$. Let $\mathcal{Q}_{\text{high}}' = \{j \in \mathcal{Q}' \mid F(j, \mathcal{Q}_j) > \alpha'\}$ and $\mathcal{Q}_{\text{low}}' = \{j \in \mathcal{Q}' | F(j, \mathcal{Q}_j) \le 2\alpha'\}$ be sets of higher and lower quality routes in $\mathcal{Q}'$. According to the definition of F, we have $w(\mathcal{Q}_{\text{low}}') \le 2\alpha' \sum_e 1 = 2m\alpha'$, where the inequality is true since an optimal solution cannot overflow an edge. Therefore, we have $w(\mathcal{Q}) \le 8m\alpha'$. In addition, since $F(j, P_j) > \alpha'$ for every $j \in \mathcal{P}$, according to [7], we have $w(\mathcal{P}) = \alpha' \sum_e L_l(e) \ge (1/4)\sqrt{m}\alpha'$. By combining the two inequalities, we get $(w(\mathcal{Q})/w(\mathcal{P})) \le 32\sqrt{m} = O(\sqrt{m})$. □

3.2.3. Rip-Up and Reroute. Rip-up and reroute is a basic routing technique and is usually combined with the negotiation technique. The negotiation-based rip-up and reroute is widely used in global routing [8, 9], track assignment [10], and detailed routing [11] and has been shown to be effective and efficient to improve the routing quality.

At each rip-up and reroute iteration, we first identify and mark a set of buses with overflowed edges or excessive routing cost (equation (9)) that need to be ripped up and rerouted. Rerouting the buses that do not overflow but have excessive routing costs can not only reduce the routing cost but also free up routing resource for other overflowed buses. Then, the marked buses are sorted in decreasing order based on the score defined as follows:

$$\begin{aligned}
S_{\text{order}}(B_i) = {} & C_1 \cdot ne_{\text{of}}(B_i) + \alpha \cdot C_w(B_i) \\
& + \beta \cdot C_s(B_i) + \gamma \cdot C_c(B_i),
\end{aligned} \tag{3}$$

where $\alpha \cdot C_w(B_i) + \beta \cdot C_s(B_i) + \gamma \cdot C_c(B_i)$ is the routing cost defined in equation (9), $ne_{\text{of}}(B_i)$ denotes the number of overflowed edges passed by bus B_i in the previous iteration, and C_1 is a user-defined parameter which is set as $\alpha + \beta + \gamma$.

In addition, the history-based cost function for each routing edge e in [8] is adopted, which is defined as

$$\text{cost}(e) = b(e) + h(e) \times p(e) + \text{vc}(e), \tag{4}$$

where $b(e)$ is the wire length cost, $h(e)$ is the history cost, $p(e)$ is the current penalty cost, $h(e) \times p(e)$ denotes the congestion cost of edge e, and $\text{vc}(e)$ is the via cost. The weight of edge e is set as $w(e) = (1/\text{cost}(e))$, and we reroute a bus by solving problem (2).

We repeat the rip-up and reroute process until there is no overflowed edge or excessive routing cost or the given maximum number of iterations is reached. Since we reroute a bus by solving the UFP (2) and the required runtimes is $O(n_{\text{bit}} \cdot |E_{P^T}|)$, the rip-up and reroute stage requires $O(n_{rb} \cdot n_{\text{bit}} \cdot |E_{P^T}|)$, where n_{rb} is the total number of buses that need to be ripped-up and rerouted.

After obtaining a 2D global routing solution, we extend the layer assignment method in [12] to map the solution from the projected plane to the original multiple layers. Note that, within each segment of a bus, we ensure that the wires of different bits are assigned to the same layer.

3.3. Track Assignment. After obtaining desired routing regions for each bus in the global routing stage, we propose a topology-aware track assignment in this subsection to allocate tracks to each segment of buses under the guidance of the global routing result. In this track assignment stage, we treat the array of all g-cells in a row or column of a routing layer as a panel, and each straight wire that passes through one or more g-cells is regarded as an iroute.

3.3.1. Initial Track Assignment. Since all bits in each bus need to be routed with the same routing topology, we handle the buses one by one to assign the tracks to each segment. For each bus, each segment consists of the set of iroutes of different bits that have the same sequence when traced from source pin to sink pin.

In order to maintain the relative order of the iroutes in each segment, our initial track assignment for each segment is as follows. First, we sort the iroutes of each segment in the same order or in the reverse order of bits. Both orders are tested because the results of each order may be different, and the best results are adopted. Then, based on the sorted order, for each iroute, we collect the valid tracks in the panels and calculate the cost of assigning the iroute to each valid track. A track is valid if the width constraint of the track is greater than or equal to the wire width of the iroute. Finally, we select a valid track with the minimum cost to accommodate the iroute.

The cost function for assigning the iroute ir to the track t is defined as

$$\begin{aligned}\mathrm{cost}(\mathrm{ir},t) = {} & \mathrm{wl}(\mathrm{ir},t) + C_2 \cdot \mathrm{ol}(\mathrm{ir},t) + C_3 \cdot \mathrm{blk}(\mathrm{ir},t) \\ & + C_4 \cdot \mathrm{cp}(\mathrm{ir},t),\end{aligned} \tag{5}$$

where $\mathrm{cost}(\mathrm{ir},t)$ is the total cost of assigning track t to iroute ir, $\mathrm{wl}(\mathrm{ir},t)$ is the wire length cost, $\mathrm{ol}(\mathrm{ir},t)$ is the overlap cost, $\mathrm{blk}(\mathrm{ir},t)$ is the blocked interval cost, $\mathrm{cp}(\mathrm{ir},t)$ is the compactness cost, and C_2, C_3, and C_4 are the user-defined constants which are set as 0.2, 1000, and 1, respectively.

The definition of wire length cost is adopted from the work [10]. Because the routing tracks may be nonuniform or even overlapping, the overlap cost of an iroute being assigned to a track is modified from the work [10], which is determined not only by the overlapping iroutes on that track but also by the overlapping iroutes on other tracks. In addition, since some tracks may only cover a partial design, the blocked interval cost is the sum of blockage cost defined in [10] and the length of the iroute ir that is not on the tracks. According to the 2018 CAD Contest at ICCAD [5], the wires of all bits in a bus should be as compact as possible. Thus, we define the compactness cost to make each bus more compact and reserve more free space for other buses.

Figure 7 illustrates the calculation of the compactness cost. We assume without loss of generality that the panels are horizontal. For the first and last segments of each bus, since the iroutes eventually need to be connected to the corresponding pins in the detailed routing stage, the compactness cost is set as the vertical distance between the iroute and the corresponding pin. For example, Figure 7(a) shows the first segment of a bus, where pin p_1 and iroute a_1 belong to the first bit and pin p_2 and iroute a_2 belong to the second bit. The compactness costs of iroute a_1 on tracks T_1, T_2, and T_3 are 0, d_1, and $d_1 + d_2$, respectively, and the compactness costs of iroute a_2 on tracks T_1, T_2, and T_3 are $d_1 + d_2$, d_2, and 0, respectively.

For the rest of the segments in the bus that do not need to connect pins, the compactness costs of the iroutes are related to the order in which the iroutes of bits are assigned. If we handle iroute a_1 before iroute a_2 in Figure 7(b), the compactness cost of each iroute is the vertical distance between the iroute and the upper boundary of the panel. Conversely, if we handle a_2 before a_1, then the compactness cost of each iroute is the vertical distance between the iroute and the lower boundary of the panel. In addition, if the iroutes of a segment are distributed in multiple panels as shown in Figure 7(c), the compactness cost of each iroute in the bottommost panel is the vertical distance between the iroute and the upper boundary of the panel, the compactness cost of each iroute in the topmost panel is the vertical distance between the iroute and the lower boundary of the panel, and the compactness cost of each iroute in the rest panels is 0.

3.3.2. Rip-Up and Reassignment. After the initial track assignment, there may be overlaps between the iroutes. Therefore, we extend the negotiation-based track assignment in the work [10] to minimize the overlaps and wire length while keeping the relative order of the iroutes in each segment.

The cost function for reassignment is defined as

$$\begin{aligned}\mathrm{cost}_{\mathrm{his}}(\mathrm{ir},t) = {} & \theta_{\mathrm{wl}} \cdot \mathrm{wl}(\mathrm{ir},t) + \theta_{\mathrm{ol}} \cdot \mathrm{ol}(\mathrm{ir},t) \\ & + \theta_{\mathrm{blk}} \cdot \mathrm{blk}(\mathrm{ir},t) + \theta_{\mathrm{cp}} \cdot \mathrm{cp}(\mathrm{ir},t) + \theta_{\mathrm{his}} \cdot \mathrm{his}(\mathrm{ir},t),\end{aligned} \tag{6}$$

where $\mathrm{wl}(\mathrm{ir},t)$, $\mathrm{ol}(\mathrm{ir},t)$, $\mathrm{blk}(\mathrm{ir},t)$, and $\mathrm{cp}(\mathrm{ir},t)$ are the same as the definition in equation (5), and $\mathrm{his}(\mathrm{ir},t)$ is the history cost from the work [10]. The user-defined parameters $\theta_{\mathrm{wl}}, \theta_{\mathrm{ol}}, \theta_{\mathrm{blk}}, \theta_{\mathrm{cp}}$, and θ_{his} are used to balance the cost components. Both θ_{wl} and θ_{cp} are initialized as 1 and gradually decreased to 0.1 as the iteration increases, and θ_{ol} is initialized as 0.1 and gradually increased to 1 as the iteration increases. Besides, θ_{blk} is a very large constant and θ_{his} is set as 1. Through the control of these parameters, we can reduce the overlaps with less wire length and compactness cost at early iterations and focus more on overlap reduction at late iterations.

In order to maintain the same topology for all bus bits, we always keep the relative order of the iroutes in each segment during the rip-up and reassignment stage. However, reassigning an iroute while keeping the relative order of iroutes in the segment may fall into a local optimum, due to the fact that the solution space is limited and we may always select the same set of iroutes or always try to assign an iroute to a small set of tracks. Therefore, in each iteration, we may rip up and reassign multiple iroutes to reduce the probability of falling into local optimum. Specifically, when an iroute has no other tracks that can be assigned to or these tracks have been tried several times by this iroute, we will also rip up and reassign some of the iroutes that adjacent to this iroute. The time complexity of reassigning an iroute is $O(n_{pt})$ at each iteration, where n_{pt} is the number of tracks in the panel that the iroute is located. Since we may rip up and reassign multiple iroutes simultaneously to reduce the probability of falling into local optimum, if we handle k iroutes simultaneously, the time complexity is $O(n_{pt}^k)$. However, since n_{pt} is not too large (tens to hundreds) and we set k as 3, the running time of this stage depends mainly on the number of iroutes that need to be reassigned.

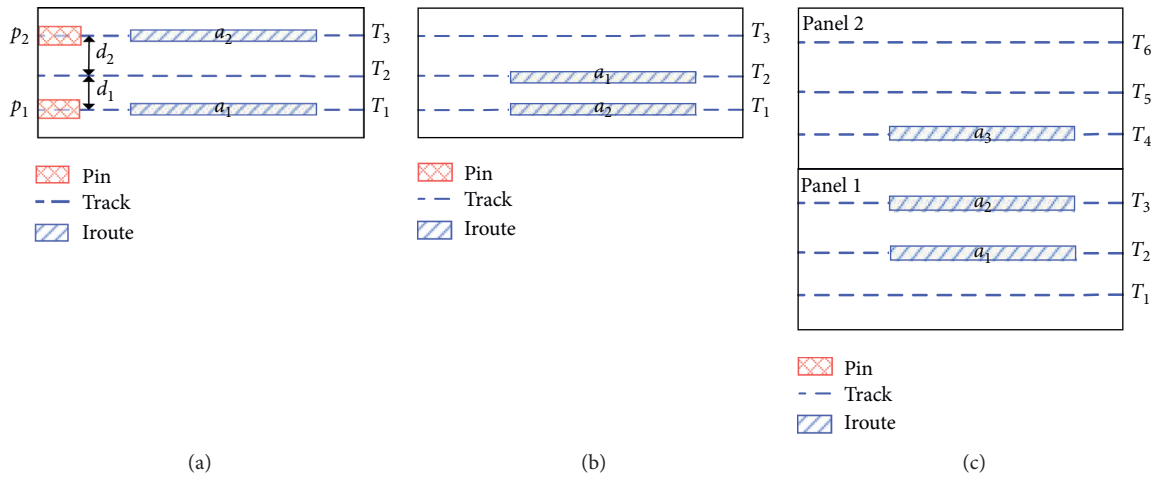


FIGURE 7: Three cases of compactness cost calculation. (a) Iroutes of the first or last segment, which eventually needs to be connected to the corresponding pins in the detailed routing stage. (b) All iroutes of a segment are in the same panel. (c) The iroutes of a segment are distributed in multiple panels.

For example, assume that iroute a_2 in Figure 7(c) needs to be reassigned. Since the relative order of the iroutes of different bits in a segment should be maintained, iroute a_2 can only be placed between iroute a_1 and iroute a_3. That is, if iroute a_1 is not reassigned to another track, then iroute a_2 cannot be reassigned. Therefore, we rip up both the iroutes a_1 and a_2, and then iroutes a_1 and a_2 can be reassigned to the tracks T_1 and T_2, respectively.

3.4. Detailed Routing. After track assignment, we need to connect the components of each bit to obtain the final routing result. A bit component is a pin or an iroute of the bit. Algorithm 2 gives the framework of our detailed routing. In Line 3, in order to preserve the same routing topology for all bus bits and honor the global routing result, the components of each bit are sorted according to the trace order of global routing paths from the source pin to the sink pin, and then we only need to connect the adjacent components of each bit one by one.

In Line 4, we adopt L-shaped [13], Z-shaped [13], and the 3-bend routing [14] to connect the adjacent components, because it is easy to control the same topology for all bus bits and is very efficient by using these predefined pattern routing. After that, we use two-stage negotiation-based rip-up and reroute to iteratively improve the solution quality in Lines 8–16. ite1 and ite2 in Line 8 indicate the maximum number of iterations for the first stage and the second stage of rip-up and reroute, respectively. In the first stage, we rip up every two adjacent components that have the overlapping wires and reroute them through the patterns routing while maintaining the same routing topology. After each iteration, we increase the history cost of the overlapped interval on a track according to the number of overlapped wires. As a result, a track with a higher history cost tends to have less chance to be routed, and the bits with alternative routes are forced to use other tracks. In the end, the bit that most needs to use this track will eventually use it.

Since the limited search space of the patterns routing and only allowing to rip up and reroute the adjacent components in the first stage rip-up and reroute, it is possible to reduce the wire overlap if some restrictions are removed. In the second stage, we use the A^* algorithm [15] to search for the paths and allow paths to be outside the global routing guide. In addition, we also allow some iroutes extracted in the track assignment stage to be removed, thus increasing the freedom of routing. For example, if we remove the second component of a bit, then we are required to find a path to connect the first component and the third component. When a part of a bit is rerouted, we check whether other bits in the bus are the same as its topology. If not, we will adjust the routing paths of other bits based on the path of that bit. We repeat the rip-up and reroute process until all the buses are routed successfully or the given maximum number of iterations is reached.

Finally, we construct a conflict graph in which each bus is regarded as a vertex, and each edge represents the conflict between two buses. We iteratively rip up the bus (vertex) with the largest degree and its associated edges until there are no conflict edges in the graph, and the final routing result is obtained.

4. Experimental Results

To evaluate our proposed bus routing algorithm, we implemented our algorithm in the C++ programming language and tested it on the benchmarks (including the hidden cases) of the 2018 CAD Contest at ICCAD on Obstacle-Aware On-Track Bus Routing [5]. Table 1 lists the benchmark statistics, where "#Layer," "#Track," "#Obstacle," "#Bus," "#Bit," and "#Pin" give the total numbers of layers, tracks, obstacles, buses, bits, and pins, respectively.

The score function in the contest [5] is adopted to evaluate the quality of bus routing results, which consists of routing cost C_r, spacing violation penalty P_s, and fail routing penalty P_f. That is,

Input: The graph $G(V, E, u)$, all bits of a bus.
Output: The set P_{best} of paths that connecting the bits.
(1) Partition the set of bits T into two disjoint sets T_1 and T_2;
(2) **for** $i = 1, 2$ **do**
(3) $\alpha_{\text{lb}} \leftarrow w_{\min}/|V|, \alpha_{\text{ub}} \leftarrow w_{\max} u_{\max}/d_{\min}$;
(4) **for** each k from $\lfloor \log \alpha_{\text{lb}} \rfloor$ to $\lceil \log \alpha_{\text{ub}} \rceil$ **do**
(5) $\alpha \leftarrow 2^k$;
(6) Sort the bits in T_i according to a nonincreasing order of w_j/d_j;
(7) **for** each $j \in T_i$ **do**
(8) **if** $\exists$ path P of bit j s.t. $F(j, P) > \alpha$ and $\forall e \in P, L_{j-1}(e) + d_j/u(e) \le 1$ **then**
(9) Route the bit on P and for $e \in P$ set $L_j(e) = L_{j-1}(e) + d_j/u(e)$;
(10) Update P_{best};
(11) **end if**
(12) **end for**
(13) **end for**
(14) **end for**

Algorithm 1: Unsplittable flow problem solving.

Input: A set of components of all bits in all buses.
Output: Final routing result.
(1) **for** each bus B_i **do**
(2) **for** each bit b_{ij} **do**
(3) Sort the components;
(4) Connect the adjacent components with the same topology;
(5) **end for**
(6) **end for**
(7) the components that wires overlap, $i \leftarrow 0$;
(8) **while** $O_c \neq \varnothing$ and $i <$ ite1 + ite2 **do**
(9) **if** $i <$ ite1 **then**
(10) Reroute1(O_c);
(11) **else**
(12) Reroute2(O_c);
(13) **end if**
(14) Update history cost andO_c;
(15) $i \leftarrow i + 1$;
(16) **end while**

Algorithm 2: Detailed routing framework.

Table 1: Statistics of the 2018 CAD Contest at ICCAD benchmarks.

Benchmark	#Layer	#Track	#Obstacle	#Bus	#Bit	#Pin
beta_1	3	49209	159	34	1260	2520
beta_2	3	49209	0	26	1262	2524
beta_3	3	22732	555108	60	665	1330
beta_4	3	22732	0	62	698	1396
beta_5	4	54150	0	6	1964	3928
final_1	3	81226	0	18	1032	2064
final_2	3	14209	0	70	1285	2570
final_3	4	21379	0	47	852	1704

Table 2: Experimental results.

Benchmark	First place				Second place				Third place				Ours				
	C_r	P_s	P_f	S	C_r	P_s	P_f	S	C_r	P_s	P_f	S	C_r	P_s	P_f	S	CPU (s)
beta_1	689	280	0	969	701	5096	0	5797	641	8744	4000	13385	812	432	0	1244	10
beta_2	515	760	0	1275	563	4904	0	5467	484	9472	2000	11956	626	224	0	850	8
beta_3	1936	0	0	1936	2024	0	0	2024	1999	1928	0	3927	1905	0	0	1905	3600
beta_4	2192	0	0	2192	2271	0	0	2271	2250	1048	0	3298	2376	184	0	2560	3600
beta_5	119	1848	0	1967	95	616	2000	2711	98	1216	2000	3314	95	0	2000	2095	45
final_1	327	830	2000	3157	367	2750	2000	5117	252	0	10000	10252	341	430	2000	2771	3600
final_2	1824	4500	8000	14324	1890	2990	8000	12880	1976	6910	0	8886	2076	1470	6000	9546	2521
final_3	2966	490	10000	13456	2678	300	2000	4978	4238	20	24000	28258	2674	540	10000	13214	3600
Normalized	1.09				2.24				4.57				1.00				

$$S = C_r + P_s + P_f. \tag{7}$$

The three parts of the score function are calculated as follows:

$$\begin{cases} C_r = \sum_{B_i \in \mathcal{B}} \left(\alpha \cdot C_w(B_i) + \beta \cdot C_s(B_i) + \gamma \cdot C_c(B_i)\right), \\ P_s = \text{number of spacing violations} \times \delta, \\ P_f = \text{number of route fail buses} \times \varepsilon, \end{cases} \tag{8}$$

where $\alpha, \beta, \gamma, \delta, \varepsilon$ are five weighting parameters given in the input data, and the values of these parameters may vary from different benchmarks. For each bus B_i, the wire length cost $C_w(B_i)$, the segment cost $C_s(B_i)$, and the compactness cost $C_c(B_i)$ are defined as follows:

$$\begin{cases} C_w(B_i) = \dfrac{\sum_j^{\text{All bits of bus } B_i} (\text{wire length of bit } j/\text{half parameter wire length of bit } j)}{\#\text{bits of bus } B_i}, \\ C_s(B_i) = \dfrac{\#\text{segment of bus } B_i}{\text{lower bound of } \#\text{segment of bus } B_i}, \\ C_c(B_i) = \dfrac{\sum_j^{\text{All bits of bus } B_i} (\text{width of segment } j/\text{lower bound width of segment } j)}{\#\text{segment of bus } B_i}. \end{cases} \tag{9}$$

Ideally, if a bus B_i is routed with the minimum wire length ($C_w(B_i) = 1$) and the minimum number of segments ($C_s(B_i) = 1$) and all segments are routed with widths close to the lower bound ($C_c(B_i) = 1$), then the routing cost C_r of the perfectly routed bus is close to $\alpha + \beta + \gamma$.

The experimental results of the top 3 teams of the 2018 CAD Contest at ICCAD [5] and ours are listed in Table 2. Our algorithm was run on a Linux workstation with a 2.40 GHz Intel Xeon CPU and 64 GB memory, and the results of top 3 teams of the contest are provided by the contest organizer. Since the binaries of top 3 teams are not available for us, we do not report their runtime. Nevertheless, the specified time for each test case is one hour according to the contest [5], and our program will be killed if the runtime exceeds the specified time.

In Table 2, the columns "First place," "Second place," "Third place," and "Ours" give the corresponding routing results generated by the first place, second place, and third place of the contest [5] and our algorithm, respectively. The latest evaluation script (eval_1.0-a8) provided by the contest [5] was used to obtain the routing cost "C_r," spacing violation penalty "P_s," fail routing penalty "P_f," and score "S." It can be seen from Table 2 that all the buses are successfully routed by our algorithm for the tested cases beta_1, beta_2, beta_3, and beta_4. Particularly, on average, our algorithm outperforms the top 3 teams by 9%, 124%, and 357% in the final scores, respectively. The experimental results show that our proposed bus routing algorithm is effective.

5. Conclusions

In this paper, we have presented an effective algorithm to solve the topology-aware bus routing problem considering the existence of both nonuniform track configuration and

obstacles. We first presented a track handling technique to unify the nonuniform routing track configuration together with obstacles. Then, we have formulated the topology-aware routing problem of single bus as UFP, which is integrated into a negotiation-based global routing problem of determining the desired routing regions for each bus. Moreover, we presented a topology-aware track assignment method which can allocate the tracks to each segment of buses regarding the guidance of the global routing result. Lastly, a detailed routing scheme has been presented to connect the segments of each bus. We have evaluated our routing results with ICCAD benchmark suites. Compared with the state-of-the-art methods, the experimental results have shown that our proposed method achieves the best overall score within the specified time.

Data Availability

The data used in this study can be accessed via http://iccad-contest.org/2018/problems.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant 2242021k30031, National Key Research and Development Project of China under Grant 2018YFB22022704, National Science Foundation of China (nos. 61977017 and 61772005), and Outstanding Youth Innovation Team Project for Universities of Shandong Province under Grant 2020KJN008.

References

[1] G. Georgiev, A. Chatterjee, and G. Iannacchione, "Exponential self-organization and moores law: measures and mechanisms," *Complexity*, vol. 2017, Article ID 8170632, 9 pages, 2017.

[2] Y. Tian and T. Watanabe, "Improved delay-matching bus routing by using multi-layers," in *Proceedings of the International Conference on Electronics Packaging and iMAPS All Asia Conference (ICEP-IAAC)*, pp. 708–713, Kyoto, Japan, April 2015.

[3] T. Yan and M. D. F. Wong, "Bsg-route: a length-matching router for general topology," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 499–505, San Jose, CA, USA, November 2008.

[4] R. Zhang, T. Pan, L. Zhu, and T. Watanabe, "A length matching routing method for disordered pins in pcb design," in *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*, Chiba, Japan, January 2015.

[5] A. Liao, H.-Y. Chang, O. Chi, and J. Wang, ICCAD 2018 CAD contest: obstacle-aware on-track bus routing, http://iccad-contest.org/2018/problems.html, 2018.

[6] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, *Handbook of Algorithms for Physical Design Automation*, Auerbach Publications, Boca Raton, FL, USA, 2008.

[7] Y. Azar and O. Regev, "Combinatorial algorithms for the unsplittable flow problem," *Algorithmica*, vol. 44, no. 1, pp. 49–66, 2006.

[8] Y.-J. Chang, Y.-T. Lee, J.-R. Gao, P.-C. Wu, and T.-C. Wang, "NTHU-route 2.0: a robust global router for modern designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1931–1944, 2010.

[9] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "NCTU-GR 2.0: multithreaded collision-aware global routing with bounded-length maze routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 709–722, 2013.

[10] M.-P. Wong, W.-H. Liu, and T.-C. Wang, "Negotiation-based track assignment considering local nets," in *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*, Macao, China, January 2016.

[11] F.-K. Sun, H. Chen, C.-Y. Chen, C.-H. Hsu, and Y.-W. Chang, "A multithreaded initial detailed routing algorithm considering global routing guides," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 81: 1–81:7, San Diego, CA, USA, November 2018.

[12] T.-H. Lee and T.-C. Wang, "Congestion-constrained layer assignment for via minimization in global routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 9, pp. 1643–1656, 2008.

[13] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Pattern routing: use and theory for increasing predictability and avoiding coupling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 7, pp. 777–790, 2002.

[14] Y. Xu, Y. Zhang, and C. Chu, "Fastroute 4.0: global router with efficient via minimization," in *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*, Yokohama, Japan, January 2009.

[15] A. Hetzel, "A sequential detailed router for huge grid graphs," in *Proceedings of the Conference on Design, Automation and Test in Europe*, Paris, France, February 1998.

Hindawi
Complexity
Volume 2021, Article ID 6618833, 7 pages
https://doi.org/10.1155/2021/6618833

Research Article

Application and Evolution for Neural Network and Signal Processing in Large-Scale Systems

Dongbao Jia,[1,2] Cunhua Li,[1] Qun Liu,[3] Qin Yu,[1] Xiangsheng Meng,[4] Zhaoman Zhong,[1] Xinxin Ban,[5] and Nizhuan Wang[1]

[1]*School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China*
[2]*MOE Key Laboratory of TianQin Project, Sun Yat-sen University, Guangzhou 510275, China*
[3]*Tianwan Nuclear Power Plant, Lianyungang 222005, China*
[4]*The First People's Hospital of Lianyungang, Lianyungang 222005, China*
[5]*School of Environmental and Chemical Engineering, Jiangsu Ocean University, Lianyungang 222005, China*

Correspondence should be addressed to Zhaoman Zhong; zhongzhaoman@163.com, Xinxin Ban; banxx@jou.edu.cn, and Nizhuan Wang; wangnizhuan1120@gmail.com

Received 10 October 2020; Revised 18 December 2020; Accepted 7 March 2021; Published 10 April 2021

Academic Editor: Dan Selisteanu

Low frequency oscillation is an important attribute of human brain activity, and the amplitude of low frequency fluctuation (ALFF) is an effective method to reflect the characteristics of low frequency oscillation, which has been widely used in the treatment of brain diseases and other fields. However, due to the low accuracy of the current analysis methods for low frequency signal extraction of ALFF, we propose the Fourier-based synchrosqueezing transform (FSST), which is often used in the field of signal processing to extract the ALFF of the low frequency power spectrum of the whole-time dimension. The low frequency characteristics of the extracted signal are compared with those of FSST and fast Fourier transform (FFT) through the resting-state data. It is clear that the signal extracted by FSST has more low frequency characteristics, which is significantly different from FFT.

1. Introduction

In recent years, with the rapid development of big data, especially artificial intelligence technology, intelligent methods have become increasingly popular in the operation and management optimization of complex large-scale systems, which often appear in ecological environments, communication networks, informatics, biology, and other disciplines. Extensive research has focused on medicine and health such as deep neural network, neuroscience, and brain science related purposes, data collection, accurate analysis, monitoring, and connecting available medical resources and healthcare services [1–6].

The brain is comprised of 10^{11} neurons and generally 10^{15} interconnections among them can handle massive information. Their shape and size are diverse, and their structures are distinct. The components of a neuron can be categorized as cell body, dendrite, and axon. Signals received from other neurons are reckoned into synapses and sent to the cell body. Once the signal entering the cell body surpasses the sustaining threshold, the neuron is burned, and the signal is transmitted to other neurons through axons [7].

The amplitude of low frequency fluctuation (ALFF) is a good measurement index of brain activation signal, which is calculated to obtain the power spectrum of brain voxel signal time series using time-frequency (TF) analysis method [8–11]. It is considered able to directly reflect the spontaneous synchronous changes of neural activity in a resting state [12, 13] and to some extent reflect the interaction and neural network connection among the relevant brain regions, which has been verified in the research of visual stimulus differences caused by eye opening and closing. Furthermore, ALFF can serve as a starting point for understanding brain diseases and can help reveal many pathological mechanisms, such as some mental diseases [14, 15]. Therefore, improving the analytical accuracy of

ALFF has a positive effect on the improvement of brain science research and the treatment of mental diseases [16–19].

Many signals such as audio signals and medical data (electrocardiogram and thoracic motion signals) can be modeled as superposition of amplitude-modulation and frequency-modulation (AM-FM) modes. Linear techniques such as wavelet transform and fast Fourier transform (FFT) are widely utilized to represent the characteristics of the signal in TF domain [20–24]. However, they all have the same limitation; that is, they cannot locate a signal with arbitrary accuracy in time and frequency. Even though an improved TF representation method called "reassignment" method (RM) was proposed, its main problem is that the reassigned transformation is no longer reversible and does not allow for the mode reconstruction. Moreover, it is difficult to achieve compatibility between the time and frequency resolutions because the frequency resolution depends on the length of the analysis window, and the analytical window length needs to be fixed according to the frequency. In particular, if the period of a signal is not constant, such as the sample signal for the object being relatively active, the analytical precision decreases [25, 26]. Therefore, FFT cannot analyze the characteristics of the signal at a certain time accurately, resulting in a large amount of signal information losses, which is unsuitable for brain science analysis particularly.

In the analysis of audio signal [27, 28], another phase-based technique called "synchrosqueezing transform" (SST) was proposed. Its purpose is similar to that of RM, which is to sharpen the time-scale representation given by wavelet transform, and has the advantage of allowing for mode retrieval. Based on the principle of wavelet-based SST, an extension of SST to the TF representation given by short-time Fourier transform called Fourier-based synchrosqueezing transform (FSST) was proposed [29], and it is proved that this method is robust to small bounded perturbations and noise [30]. Moreover, some researches proposed improving the existing FSST by using higher-order amplitude and phase approximation to calculate more accurate instantaneous frequency estimates of the modal components of the signal. It is achieving a perfect concentration and reconstruction in a wider range of AM-FM modes, and most of the real signals are composed of strongly modulated AM-FM modes, such as chirps involved in radar [31], speech processing [32], or gravitational waves [9, 11, 33]. As a consequence, this technique provides a highly concentrated TF representation for a wide variety of AM-FM multicomponent signals and enables the reconstruction of their modes with high accuracy; we hypothesized that FSST can be used in the analysis of time series of brain science and to improve the analytical accuracy.

In the present study, we propose and attempt the application of FSST to brain wave analysis and compare the results obtained by FFT and FSST analysis using data with no task resting state. Based on learning and improving the previous ALFF algorithm, the ALFF in the overall time domain is recalculated. According to the comparison with traditional ALFF values, we demonstrate that FSST is a highly precise method, which yields marked improvements in the reconstruction of TF signal. This technique is effective for analyzing and visualizing ALFF values and changes in brain waves.

2. Materials and Methods

The FFT as a Fourier-related transform has been commonly used for analyzing organic signals, such as the brain wave signal. The Fourier transform can be represented as follows:

$$X(nf1) = \int_{-\infty}^{\infty} x(t)e^{-\inf_1 t}\mathrm{dt}, \tag{1}$$

where t is the analysis window length ($f_1 = 1/t$, n is a positive integer). Equation (1) is solved for determining the Fourier coefficients. As the Fourier transform is used for analyzing a completely periodic signal in an analysis window t, the calculated frequencies ($nf1$) depend on the window length t, and errors frequently occur in the analysis of nonharmonic signal frequencies. Moreover, if changes in the frequency of the sample signal for the object are relatively active over a short duration, a decrease in the length of the analysis window to increase the time resolution also decreases the frequency resolution; the resulting analytical precision decreases.

As we all know, it is not appropriate to describe the nonstationary signal with time-domain or frequency-domain representation. Therefore, the short-time Fourier transform (STFT) is introduced and defined as follows:

$$V_f^g(t,\eta) = \int_R f(\tau)g^*(\tau - t)e^{-2i\pi\eta(\tau - t)}d\tau\ , \tag{2}$$

where g* is the complex conjugate of g. Then, the spectrum corresponds to $|V_f^g(t,\eta)|^2$, and the original signal f can be extracted from the STFT under the condition that g does not disappear and is continuously at 0:

$$f(t) = \frac{1}{g^*(0)}\int_R V_f^g(t,\eta)d\eta\ . \tag{3}$$

If f is analytic, which means $\eta \le 0$, then $\widehat{f}(\eta) = 0$; the integral in (3) only happens on R_+.

Synchrosqueezing transform can separate and adjust the signals of different modes by redistributing coefficients in series or frequency to avoid the uncertainty of linear transformation. It can analyze and compress the low frequency amplitude, obtain more accurate high-power signals, and find active points in brain regions. The signal can be expressed clearly while maintaining the reversibility of the signal.

The Fourier-based synchrosqueezing transform (FSST) technique is a newly proposed analytical method that has some advantages, as evidenced by the numerical processing of synthetic and observed gravitational wave signals [34, 35]. Time t and frequency η are predicted by sharpening the blurred STFT representation using the following instantaneous frequency of FSST:

Table 1: Relevant information of echo planar imaging and fast gradient echo image.

	Slices	Thickness/ gap	In-plane resolution	Repetition time (ms)	Echo time (ms)	Field of view	Turning angle	Inversion time
Echo planar imaging	33 axial	3.5/0.7 mm	64*64	2000	30	200*200 mm^2	90°	—
Fast gradient echo image	128 sagittal	1.33/0 mm	256*192	2530	3.39	256*256 mm^2	7°	1100 ms

$$\widehat{\omega_f}(t,\eta) = \frac{1}{2\pi}\partial_t \arg\left\{V_f^g(t,\eta)\right\}, \tag{4}$$

where ∂_t is the partial derivative with respect to t. Indeed, $V_f^g(t,\eta)$ is redistributed to a new location $(t, \widehat{\omega_f}(t,\eta))$ using the following synchrosqueezing operator:

$$T_f^{g,\gamma}(t,\omega) = \frac{1}{g^*(0)}\int_{\left\{\eta,\left|V_f^g(t,\eta)\right|>\gamma\right\}} V_f^g(t,\eta)\delta\left(\omega - \widehat{\omega}_f(t,\eta)\right)d\eta, \tag{5}$$

where γ is the threshold. By integrating $T_f^{g,\gamma}(t,\eta)$ near the corresponding ridge $(t, \phi_k'(t))$, the k^{th} mode is reconstructed approximately, since the coefficients are redistributed along the frequency axis, and FSST maintains the nature of causality:

$$f_k(t) \approx \int_{\{\omega,|\omega-\varphi_k(t)|<d\}} T_f^{g,\gamma}(t,\omega)d\omega\ , \tag{6}$$

where $\varphi_k(t)$ is an estimate of $\phi_k'(t)$ and the parameter d enables compensating for the imprecise approximation $\varphi_k(t)$ of $\phi_k'(t)$ and the error made by estimating the IF by means of $\widehat{\omega_f}(t,\eta)$. In addition, the approximation $\varphi_k(t)$ must be calculated before retrieving mode f_k. As a consequence, a commonly used technique that is based on ridge extraction predicting $T_f^{g,\gamma}$ and K is known [33]. Therefore, FSST as a highly precise frequency analysis method can be used to analyze brain waves and improve the analytical precision in the present study.

2.1. Experiment. Forty-seven participants in their 20s–40s with no mental illness history participated in the present study of resting state. All participants provided written consent to participate, and ethical approval was obtained from the institutional ethics committee [36, 37]. For avoiding the influence of outside equipment, the experiment was conducted in an anechoic chamber with lights off.

In functional imaging, echo planar imaging and fast gradient echo image were obtained by three-dimensional T1-weighted magnetization with the relevant information shown in Table 1, respectively [38–40].

All of forty-seven participants' closed-eye sample data were used in the resting experiment. Each condition consists of 240 functional volumes; in terms of data selection, we discard the original 10 point volumes and reserve 120 volumes.

The improvement of the algorithm is mainly reflected in the calculation and selection of power spectrum; in terms of analytical conditions, as the brain activity is the most obvious and active in this piece of data, FFT and FSST both select the average value in the time series with the frequency band of 0.01–0.1 Hz and the data length of 20–420 s in the total length of 9 minutes. The analysis window length is 2*ABS (data)/sample length. Furthermore, FFT and FSST algorithms are used to extract ALFF values and conduct comparative analysis; the coefficient values of the corresponding ALFF are extracted as follows (Table 2).

3. Results and Discussion

The experimental data of each participant were extracted twice. We divided these two groups of data into Test1 and Test2; FFT and FSST are used to analyze the two groups of data. Results of participant 32 as an example are shown in Figure 1 since it is nonspecific, which are Test1 FSST, Test1 FFT, Test2 FSST, and Test2 FFT, respectively. Each group of images shows 12 cross sections of the brain, and the corresponding position of each cross section is shown in the longitudinal view of brain area in the lower right corner of each group of images.

Although the high amplitude of FSST and FFT is only about 3000 in the resting state, the difference between FSST and FFT is obvious. From the point of view of images, FSST detects the results that cannot be resolved by FFT and their mean coefficients, which proves that FSST is more accurate and sensitive than FFT in data analysis and image extraction and highlighting. In particular, FSST highlights the superior frontal fornix, rectus, and other areas, which are valuable for the scientific research of brain activity and mental disease. Therefore, the difference between FSST and FFT is large, whose accuracy is also higher.

Since the amplitude of the low frequency band is used in this experiment, the average sum of these values should be carried out in FFT analysis of these multiple detection points, and then the frequency power spectrum is obtained by judging the ratio of the low frequency band (0.01–0.1 Hz) to each frequency band; the obtained frequency power spectrum is analyzed to determine whether these points conform to the low frequency amplitude attribute. To further analyze the significance of these signals, we introduce the mean correlation coefficient to correlate the extracted signals and calculate the average value, as shown in the figure below.

We can clearly see that, in both the mean coefficient and the low frequency ratio of the two groups of data, the multiple detection parts of FSST are indeed better than those

Table 2: Coefficient values of ALFF for FFT and FSST in resting-state experiment.

Participant	FFT	FSST	Participant	FFT	FSST
1	0.85	0.81	25	0.85	0.90
2	0.92	0.87	26	0.89	0.82
3	0.94	0.90	27	0.89	0.85
4	0.96	0.91	28	0.90	0.89
5	0.91	0.86	29	0.90	0.82
6	0.95	0.91	30	0.95	0.89
7	0.92	0.88	31	0.93	0.94
8	0.84	0.74	32	0.87	0.77
9	0.87	0.85	33	0.52	0.47
10	0.94	0.93	34	0.84	0.80
11	0.82	0.75	35	0.91	0.86
12	0.95	0.92	36	0.86	0.74
13	0.81	0.85	37	0.92	0.89
14	0.91	0.89	38	0.90	0.87
15	0.94	0.91	39	0.93	0.93
16	0.86	0.70	40	0.92	0.91
17	0.87	0.78	41	0.79	0.70
18	0.95	0.92	42	0.88	0.83
19	0.92	0.90	43	0.90	0.87
20	0.94	0.87	44	0.92	0.92
21	0.94	0.90	45	0.95	0.93
22	0.97	0.95	46	0.93	0.92
23	0.90	0.89	47	0.96	0.93
24	0.95	0.94	**Mean**	**0.90**	**0.86**

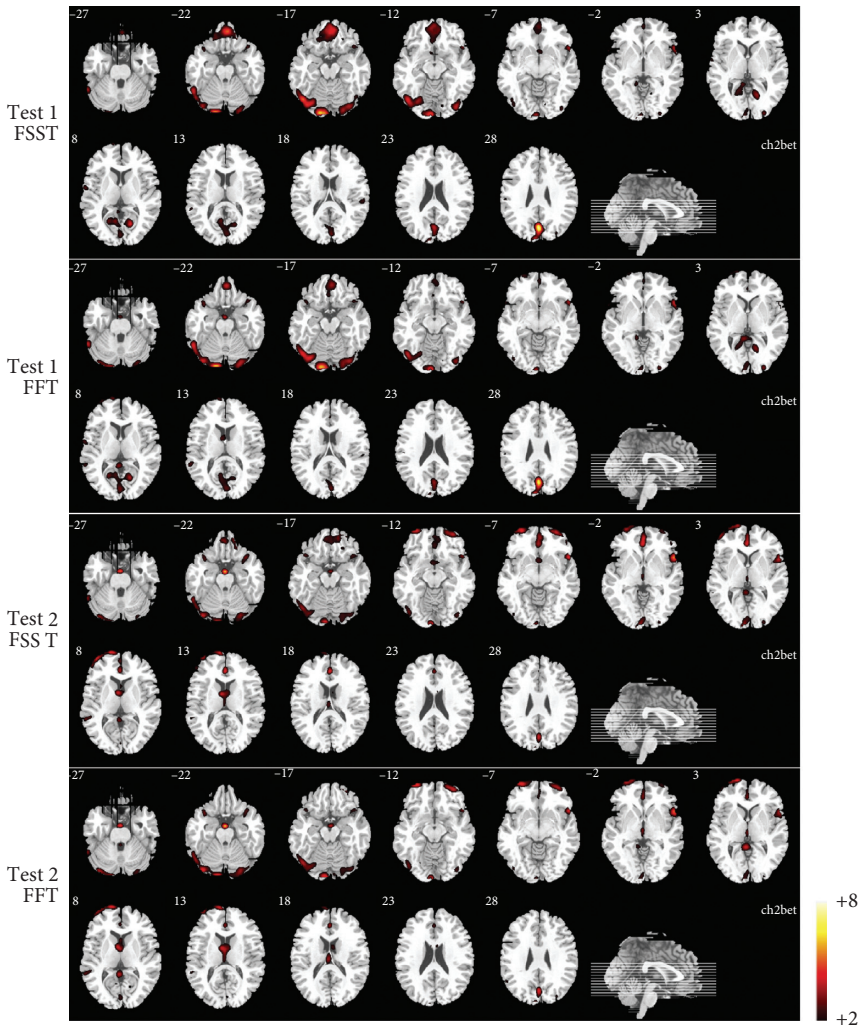


Figure 1: Analysis results of 2 groups of data from participant 33 using FSST and FFT technique.

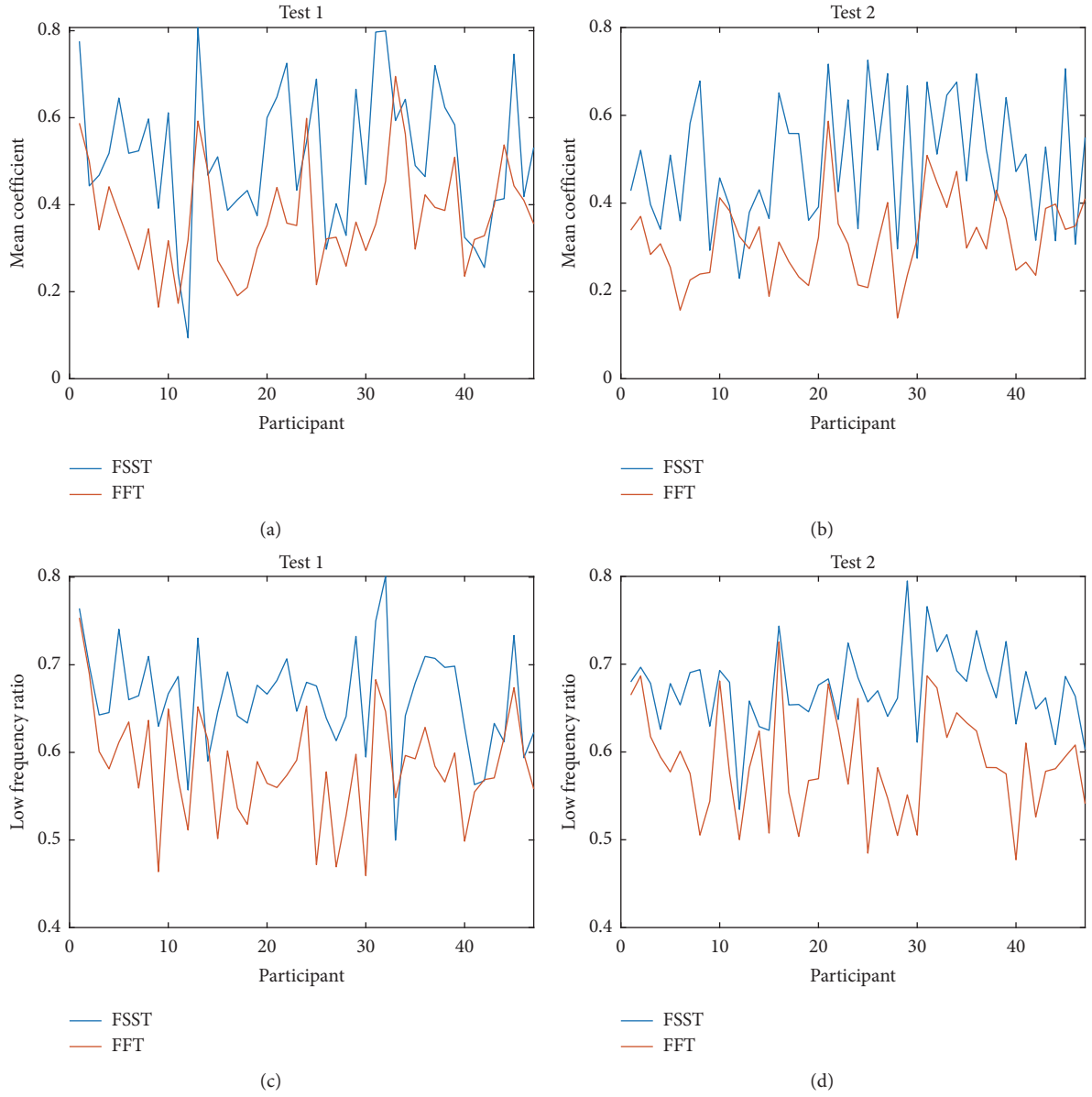


Figure 2: Mean coefficient and low frequency ratio of each participant in low frequency band using FSST and FFT.

of FFT, as shown in Figure 2. Since the significance of the low frequency coefficient can reflect the significance of the ALFF index and the difference of the low frequency amplitude is the reason for the difference of the low frequency coefficient, the low frequency amplitude value targeted by this experiment can show the characteristics of ALFF. Therefore, it is also proved that FSST is clearly superior to FFT in the analysis of ALFF value, which has stronger low frequency characteristics and more obvious ability to extract low frequency characteristics.

Figure 3 shows the area with P value less than 0.05 after the examination. Set 18 adjacent voxels (edge connection, the default value of SPM) and voxel number of 200 as the cluster size parameter, perform cluster reading on these regions, and the region with voxel greater than 70 is selected from the cluster for display, since the number of regions is too large. The positive value is the region with significant difference between FSST and FFT. Negative values represent areas where there is a significant difference between FFT and FSST. As shown in Figure 2, the FSST and FFT also show significant differences in Figure 3.

Based on the analysis and the performance of the actual effect, the significance of the algorithm improvement is proved. The performance of ALFF index has been greatly improved which promotes the application of ALFF and its value.

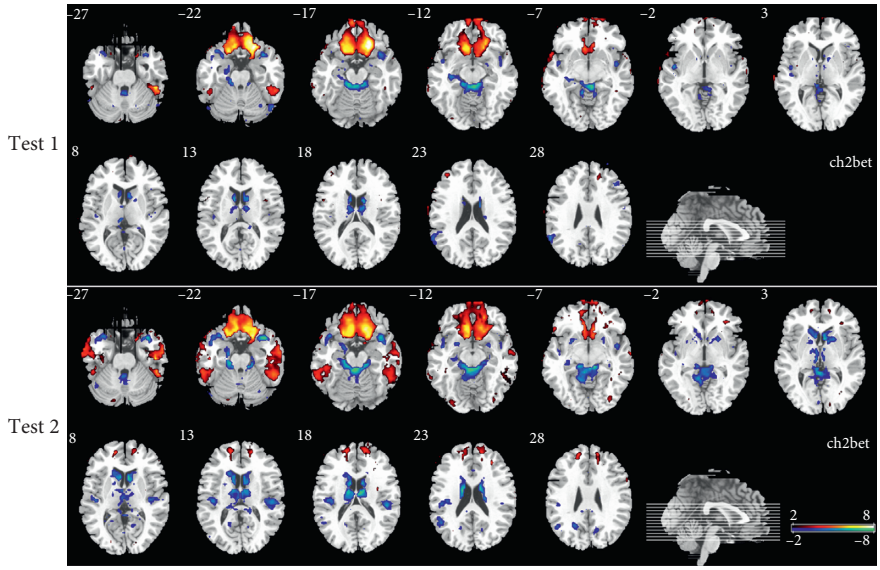


Figure 3: Difference results of FSST and FFT in Test1 and Test2.

4. Conclusions

At present, the research on human brain occupational plasticity has been developed rapidly, but the research on ALFF needs to be further improved. Because of the complexity and variety of EEG signals, extracting the information of the signals in time series can effectively reduce the signal ignorance caused by frequency band concentration.

Based on the previous research, this paper proposes the FSST algorithm to improve the resolution and enhance the analytical accuracy of ALFF, aiming to improve the visualization effect of ALFF and the extraction effect of low frequency signals. From the experimental results, it is proved that there is a significant difference between the overall analytical effects of FSST and FFT in resting state, and the extraction precision of FSST is significantly higher, which can reflect the TF structure characteristics of brain waves clearly.

In addition, FSST can detect the changes of brain dynamics in time series. After the processing, the TF map of the signal can be recovered to a greater extent. Based on the accuracy of the recovered signal and the characteristics of low frequency attributes, it is proved that FSST algorithm is feasible for the detection and analysis of ALFF in the time domain. In the future, we will verify the applicability of FSST to ALFF in task-related experiments.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grants 19KJB160001 and 19KJB520004, the Lianyungang city Haiyan project under Grant 2019-QD-004, the MOE Key Laboratory of TianQin Project, Sun Yat-sen University, and the Open Fund Project of Jiangsu Institute of Marine Resources Development under Grant JSIMR202018. The authors would like to thank Qin Yu for his contribution at the early phase of this work.

References

[1] R. Xu, N. Jiang, C. Lin et al., "Enhanced low-latency detection of motor intention from EEG for closed-loop brain-computer interface application," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 288–296, 2014.

[2] J. Kevric and A. Subasi, "Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system," *Biomedical Signal Processing and Control*, vol. 31, pp. 398–406, 2017.

[3] S. A. Sprague, M. T. McBee, and E. W. Sellers, "The effects of working memory on brain-computer interface performance," *Clinical Neurophysiology*, vol. 127, no. 2, pp. 1331–1341, 2016.

[4] S. P. Levine, J. E. Huggins, S. L. BeMent et al., "A direct brain interface based on event-related potentials," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 180–185, 2000.

[5] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, p. 1. In press, 2020.

[6] J. Li, X. J. Duan, Q. Cui et al., "More than just statics: temporal dynamics of intrinsic brain activity predicts the suicidal ideation in depressed patients," *Psychological Medicine*, vol. 49, no. 5, pp. 1–9, 2018.

[7] S. Gao, M. Zhou, Y. Wang, J Cheng, H Yachi, and J Wang, "Dendritic neuron model with effective learning algorithms

for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[8] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 233–246, 2020.

[9] D. Jia, K. Yanagisawa, Y. Ono et al., "Multiwindow non-harmonic analysis method for gravitational waves," *IEEE Access*, vol. 6, pp. 48645–48655, 2018.

[10] L. Yang, Y. Yan, Y. H. Wang et al., "Gradual disturbances of the amplitude of low-frequency fluctuations (ALFF) and fractional ALFF in Alzheimer spectrum," *Frontiers in Neuroscience*, vol. 12975 pages, 2018.

[11] D. Jia, K. Yanagisawa, M. Hasegawa et al., "Time-frequency-based non-harmonic analysis to reduce line-noise impact for LIGO observation system," *Astronomy and Computing*, vol. 25, pp. 238–246, 2018.

[12] Y. R. Wang, S. C. Gao, M. C. Zhou et al., "A multi-layered gravitational search algorithm for function optimization and real-world problems," *IEEE/CAA Journal of Automatica Sinica*, In press, 2020.

[13] D. Jia, H. Dai, Y. Takashima et al., "EEG processing in internet of medical things using non-harmonic analysis: application and evolution for SSVEP responses," *IEEE Access*, vol. 7, pp. 11318–11327, 2019.

[14] Q. Cui, W. Sheng, Y. Chen et al., "Dynamic changes of amplitude of low-frequency fluctuations in patients with generalized anxiety disorder," *Human Brain Mapping*, vol. 41, no. 6, pp. 1667–1676, 2020.

[15] W. Liao, H. Chen, J. Li et al., "Endless fluctuations: temporal dynamics of the amplitude of low frequency fluctuations," *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2523–2532, 2019.

[16] G. Rodríguez-Bermúdez, P. J. García-Laencina, J. Roca-González, and J. Roca-Dorda, "Efficient feature selection and linear discrimination of EEG signals," *Neurocomputing*, vol. 115, pp. 161–165, 2013.

[17] J. Atkinson and D. Campos, "Improving BCI-based emotion recognition by combining EEG feature selection and kernel classifiers," *Expert Systems with Applications*, vol. 47, pp. 35–41, 2016.

[18] Linux.com. Who needs the internet of things? https://www.linux.com/news/who-needs-internetthings.

[19] D. B. Jia, Y. Fujishita, C. H. Li et al., "Validation of large-scale classification problem in dendritic neuron model using particle antagonism mechanism," *Electronics*, vol. 9, no. 5, 2020.

[20] H. Shen, L. Zhao, Y. Bian, and L. Xiao, "Research on SSVEP-based controlling system of multi-DoF manipulator," *Advances in Neural Networks—ISNN 2009*, Springer, Berlin, Germany, pp. 171–177, 2009.

[21] Z. Iscan and V. V. Nikulin, "Steady state visual evoked potential (SSVEP) based brain-computer interface (BCI) performance under different perturbations," *PLoS One*, vol. 13, no. 1, Article ID e0191673, 2018.

[22] A. M. Dreyer and C. S. Herrmann, "Frequency-modulated steady-state visual evoked potentials: a new stimulation method for brain-computer interfaces," *Journal of Neuroscience Methods*, vol. 241, pp. 1–9, 2015.

[23] J. Artieda, M. Valencia, M. Alegre, O. Olaziregi, E. Urrestarazu, and J. Iriarte, "Potentials evoked by chirp-modulated tones: a new technique to evaluate oscillatory activity in the auditory pathway," *Clinical Neurophysiology*, vol. 115, no. 3, pp. 699–709, 2004.

[24] R. Ku, A. Duszyk, P. Milanowski et al., "On the quantification of SSVEP frequency responses in human EEG in realistic BCI conditions," *PLoS One*, vol. 8, no. 10, Article ID e77536, 2013.

[25] C.-M. Huang, Y.-F. Chen, S. Xu, and H. Zhou, "The vehicular social network (VSN)-Based sharing of downloaded geo data using the credit-based clustering scheme," *IEEE Access*, vol. 6, pp. 58254–58271, 2018.

[26] D. J. A. McKechan, C. Robinson, and B. S. Sathyaprakash, "A tapering window for time-domain templates and simulated signals in the detection of gravitational waves from coalescing compact binaries," *Classical Quantum Gravity*, vol. 27, no. 8, Article ID 084020, 2010.

[27] I. Daubechies and S. Maes, "A nonlinear squeezing of the continuous wavelet transform based on auditory nerve models," *Wavelets in Medicine and Biology*, pp. 527–546, Routledge, Abingdon, UK, 1996.

[28] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243–261, 2011.

[29] G. Thakur and H.-T. Wu, "Synchrosqueezing-based recovery of instantaneous frequency from nonuniform samples," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 5, pp. 2078–2095, 2011.

[30] G. Thakur, E. Brevdo, N. S. Fučkar, and H.-T. Wu, "The synchrosqueezing algorithm for time-varying spectral analysis: robustness properties and new paleoclimate applications," *Signal Processing*, vol. 93, no. 5, pp. 1079–1094, 2013.

[31] M. Skolnik, *Radar Handbook, Technology and Engineering*, McGraw-Hill Education, New York, NY, USA, 2008.

[32] J. W. Pitton, L. E. Atlas, and P. J. Loughlin, "Applications of positive time-frequency distributions to speech processing," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 554–566, 1994.

[33] D.-H. Pham and S. Meignen, "High-order synchrosqueezing transform for multicomponent signals analysis-with an application to gravitational-wave signal," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3168–3178, 2017.

[34] S. Klimenko and G. Mitselmakher, "A wavelet method for detection of gravitational wave bursts," *Classical and Quantum Gravity*, vol. 21, no. 20, pp. S1819–S1830, 2004.

[35] B. P. Abbott, R. Abbott, T. D. Abbott et al., "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, Article ID 061102, 2016.

[36] Ethics Committee, University of Jiangsu, "Ethical review of research for non-medical purposes," University of Jiangsu, Zhenjiang, China, 2020, https://kjc.ujs.edu.cn/info/1149/2963.htm.

[37] G. Garcia, "High frequency SSVEPs for BCI Applications," in *Proceedings of the Brain Computer Interfaces HCI Games Workshop*, Florence, Italy, January 2008.

[38] D. Liu, Z. Dong, X. Zuo, J. Wang, and Y. Zang, "Eyes-open/eyes-closed dataset sharing for reproducibility evaluation of resting state fMRI data analysis methods," *Neuroinformatics*, vol. 11, no. 4, pp. 469–476, 2013.

[39] A. Dutta, S. McKie, and J. F. W. Deakin, "Resting state networks in major depressive disorder," *Psychiatry Research: Neuroimaging*, vol. 224, no. 3, pp. 139–151, 2014.

[40] C.-G. Yan, X.-D. Wang, X.-N. Zuo, and Y.-F. Zang, "DPABI: data processing & analysis for (Resting-State) brain imaging," *Neuroinformatics*, vol. 14, no. 3, pp. 339–351, 2016.

Hindawi
Complexity
Volume 2021, Article ID 8826833, 19 pages
https://doi.org/10.1155/2021/8826833

Research Article

A Time-Aware Hybrid Approach for Intelligent Recommendation Systems for Individual and Group Users

Zhao Huang [1,2] **and Pavel Stakhiyevich** [2]

[1]*Key Laboratory of Modern Teaching Technology, Shaanxi Normal University, Xi'an, China*
[2]*School of Computer Science, Shaanxi Normal University, Xi'an, China*

Correspondence should be addressed to Pavel Stakhiyevich; t329023@gmail.com

Received 21 September 2020; Revised 26 October 2020; Accepted 19 February 2021; Published 28 February 2021

Academic Editor: Rui Wang

Although personal and group recommendation systems have been quickly developed recently, challenges and limitations still exist. In particular, users constantly explore new items and change their preferences throughout time, which causes difficulties in building accurate user profiles and providing precise recommendation outcomes. In this context, this study addresses the time awareness of the user preferences and proposes a hybrid recommendation approach for both individual and group recommendations to better meet the user preference changes and thus improve the recommendation performance. The experimental results show that the proposed approach outperforms several baseline algorithms in terms of precision, recall, novelty, and diversity, in both personal and group recommendations. Moreover, it is clear that the recommendation performance can be largely improved by capturing the user preference changes in the study. These findings are beneficial for increasing the understanding of the user dynamic preference changes in building more precise user profiles and expanding the knowledge of developing more effective and efficient recommendation systems.

1. Introduction

The fast proliferation of online information increases the users' difficulties in finding target information, services, and products on the Internet. Recommendation systems act as a filtering service to fight against the information overload [1], with the clear purpose of identifying precise positioning of the information targets, as well as offering efficient resource utilization [2]. Their main functionality is to provide the users with recommendations that are more in line with the user personal preferences [3]. Moreover, recommendation systems can also increase business revenue [4], improve business efficiency [5], and strengthen the users' loyalty toward business [6]. Currently, recommendation systems have been widely used in a variety of application domains, such as entertainment, commerce, and social networks, and many popular web platforms, including YouTube, Amazon, Spotify, and Facebook, largely leverage recommendation systems and technologies in their business to increase business profits and effectiveness [4], and promote the user satisfaction [7] and loyalty [8].

Recommendation systems are increasingly drawing the attention of the practitioners and the academics. Evidence from prior research shows that in addition to relevance, novelty and diversity are also important factors that need to be addressed in assessing recommendation systems [9]. Indeed, novel and diverse recommendations could not only help the users find relevant information and services but also support the users in discovering new items [10]. Furthermore, the novelty and the diversity of the recommendation systems can cover the shortages of the recommendation systems. For example, the problem of long-tail items (e.g., less popular and newly added items) can be effectively addressed by diverse (e.g., with variant and wide-ranging features) recommendations [11]. Moreover, the influence of repeated recommended items on user satisfaction and business sales can be alleviated by considering the novelty of recommendations [12, 13]. Therefore, it is important to

focus on relevance, novelty, and diversity in recommendation systems.

Recommendation systems can be generally categorized into personal recommendation systems and group recommendation systems according to their target range and the quantity of users [14]. Personal recommendation systems aim to provide a single user with relevant product or service recommendations, while group recommendation systems recommend items for a group of users [15]. The major differences between these two categories can be highlighted in terms of system design, user interaction, and business purposes. Regarding system design, personal recommendation systems usually use collaborative filtering and content-based methods to provide single users with information to aid them in information seeking [16], whereas group recommendation systems normally utilize aggregation techniques to offer recommendations for the groups of users [17]. For user interaction, personal recommendation systems involve explicit and implicit user-item interactions where each user interacts with items separately, whereas group recommendation systems involve interactions between groups of users and items where each user is represented as a part of a group. For business purposes, personal recommendation systems are mainly dedicated to support and fulfill the individual user's goals, thereby increasing the profit, satisfaction, and loyalty of the individual users. At the same time, group recommendations are normally trying to assist and accomplish the group's goals to increase the profit, satisfaction, and loyalty of the groups of users.

Although both personal and group recommendations have been quickly developed, challenges such as dynamic user preference changes, precise user profile establishment, data sparsity and recommendation diversity still exist. In particular, the issue of user preference changes over time has become one of the most challenging tasks because it can cause difficulties in building accurate user profiles within recommendation systems and has a considerable impact on recommendation performance. More importantly, user preference changes are complex and are usually grouped into long-term and short-term preferences [18, 19]. Long-term preferences can be more stable and change slowly over a certain period. Short-term preferences can be quickly influenced by user instant demands and recent interests. Therefore, to better understand the changes in the user preferences over time, build more accurate user profiles and improve the recommendation performance. Also, the time awareness of the user preferences for individual and group recommendations needs to be explored further.

Some studies examine the user preference changes (e.g., [20, 21]) and explain the role of long-term and short-term preferences in recommendations (e.g., [22, 23]), but there is a limited focus on the user preference dynamics. It can be argued that without drawing enough attention to the user preferences dynamics, the recommendation performance could be limited or dropped as recommendations that have been relevant in the past might not meet users' preferences at present because the users have changed their preferences [24, 25]. Even those rare studies that investigate dynamics in the user preferences (e.g., [26]) do not offer a systematic understanding of the user preferences changing and lack the analysis of relevant effects on recommendations. More importantly, there is no study that particularly addresses time awareness in the recommendations for both individuals and groups of users. It may be arguable that this lack of understanding may hinder the development of effective and efficient recommendation systems. Hence, this study primarily investigates the user preference changes over time to fulfill its promise. More specifically, the following research questions are investigated: (1) What are the effects of the user preference changes on the individual user and the group recommendations? (2) How do user groups affect the precision, recall, novelty, and diversity of group recommendations? To carry out the research, an experimental study is conducted by proposing a time-aware hybrid approach to generate recommendations for individuals and groups of users. The proposed approach consists of collaborative filtering, a content-based method, and the method that aggregates personal recommendations in this study. Collaborative filtering is employed for candidate item selection. The content-based method is used for building the user profiles as well as providing top personal recommendations. The aggregation method for personal recommendations is used for group recommendations.

This study contributes to the recommendation systems' research by developing a time-aware hybrid recommendation approach for both individual and group users in which the user preferences change over time and are importantly addressed to improve the recommendation performance. This approach will help developers increase their understanding of capturing the user dynamic preference changes in building more precise user profiles as well as strengthening the knowledge of the user profile modeling for personal and group recommendations. Our study can also help scholars focus on multiple metrics, including precision, recall, novelty, and diversity, to evaluate the recommendations' outcomes, providing deeper insights into measuring the efficiency of the recommendation systems.

The rest of this paper is organized as follows. The section of related work describes a variety of approaches for personal and group recommendation, and the importance of the user preference change is addressed. In the research methodology section, we introduce a time-aware hybrid recommendation approach used in this study. The section of experimental study identifies evaluation metrics and describes the design of personal and group recommendations. Then, the detailed experimental results are described and discussed in the results and discussion section. Finally, our conclusions, implications, limitations, and future work are described in the conclusion section.

2. Related Work

2.1. Personal Recommendations. Personal recommendation systems aim to provide a single user with relevant product or service recommendations. They support individual users in the decision-making process, and help to achieve users' goals and fulfill their needs. A variety of recommendation techniques have been used for developing personal

recommendation systems, such as collaborative filtering, content-based filtering, and knowledge-based and demographic recommendation systems. However, most of these techniques have challenges and limitations, including cold-start [27], data sparsity [28], and limited content analysis [29] problems. To overcome these problems and limitations, hybrid recommendation systems are popularly used. In recent years, hybrid recommendation systems that combine two or more recommendation techniques have mainly been used for personal recommendations. More importantly, hybrid personal recommendation systems could be further improved by exploring additional information, such as user-generated content and supplementary user information and context item information.

User-generated content, such as user reviews, item description tags, and ranking, can help recommendation systems better understand the user preferences and improve the recommendation performance. For example, Wang et al. [30] developed a hybrid collaborative filtering method that combines preliminary recommendations with the sentiment analysis of the user reviews to increase the accuracy of recommendations. Likewise, Qian et al. [31] proposed an emotion-aware recommendation system based on the hybrid information fusion. The results show that there is a significant improvement in the recommendation performance by using more user-generated information.

Furthermore, the performance of the hybrid personal recommendation systems can be promoted by acquiring supplementary user information and contextual item information [32–34]. The common techniques that are used for acquiring this information include ontology and deep learning. Tarus et al. [32] used a hybrid recommendation approach that consists of ontology and sequential pattern mining techniques. The ontology is used to depict the knowledge about user interests and item features, which can help their proposed approach to alleviate the cold-start and data sparsity problems. Likewise, sequential pattern mining is used for identifying the user historical sequential learning patterns to increase the recommendation performance and improve the accuracy of predictions. Similarly, Kermany and Alizadeh [33] carried out an ontology-based study, developing a hybrid recommendation system that employs fuzzy multicriteria collaborative filtering and item-based ontological semantic filtering approaches. This study addresses users' demographic information as external information sources to improve the performance of the recommendation system. In addition, Kim et al. [34] conducted a study that uses the convolutional neural network with probabilistic matrix factorization to capture the contextual information of the items. The findings confirm the effectiveness of using convolutional neural networks in capturing contextual information from item descriptions, showing the higher quality of recommendations.

2.2. Group Recommendations. A group recommendation system is another popular recommendation system that focuses on recommendations to the groups of users. To provide more precise group recommendations, the user model aggregations are important approaches that integrate the individual user preference profiles as the group profile to generate relevant recommendations [17, 35]. These approaches, such as the average aggregations and least misery aggregations, primarily focus on user profiles, leveraging user individual preferences to group preferences to generate group recommendations. However, simple user model aggregation methods for group recommendations could lead to lower performance because the users could have contradictory preferences and the number of users in a group could be large. To cope with contradictive preferences among users in a group, Guo et al. [36] developed a group recommendation approach in which the group recommendations process is transformed into a multicriteria decision-making process (MCDM). This method can better address the contradicting preferences in the group and alleviate the cold-start problem in the group recommendation. The concurrent work of Guo et al. [37] has also considered a similar approach based on the user model aggregations. More specifically, the authors explored the heterogeneity among users' preferences and aggregated the predicted preference relations into the group profile by utilizing the Borda voting rule. The results prove that utilizing preference relations optimizes group profile modeling and improves the efficiency of the recommendations. In addition, to address a large number of users in a group, Seo et al. [35] introduced a user model aggregation method that takes into account deviations for group recommendations to improve the recommendation performance even in cases with a large number of users in a group. Overall, model aggregation approaches show their efficiency in recommendations for the users with contradictive preferences and large groups of users. However, the main problem of model aggregation methods lies in the difficulty of integrating them into hybrid recommendation systems, which could limit their performance in group recommendations.

Another approach, which is importantly used for group recommendations, is the aggregation of personal recommendations [38]. This approach primarily generates recommendations to each user of a group and then converts generated personal recommendations into group recommendations. For example, Villavicencio et al. [39] used the multilateral Monotonic Concession Protocol (MCP) to combine individual recommendations provided for each user in a group into group recommendations. The authors proposed an extension of a multiagent approach based on a negotiation technique that improves the quality of the group recommendations. More importantly, the personal recommendation aggregation approaches can be better integrated into hybrid recommendation systems to achieve desirable recommendation outcomes. For example, Kassak et al. [40] combined collaborative and content-based methods and converted the recommendations for individual users into group recommendations by utilizing the conjunctive aggregation function. By doing so, it can better address group conflict preferences and improve the quality of group recommendations. Similarly, Pessemier et al. [41] presented a hybrid recommendation system that combines individual recommendations into group recommendations using a

two-step aggregation method. In the first step, the average without misery (AvgWM) method is used to generate a list of group recommendations. The concept of average without misery method is to find the optimal decision for the group without offending any participant with this decision In the second step, the users give feedback to the generated list and select their final favorite recommendations.

2.3. User Preferences Changes. The users change their preferences as time passes [20]. Rafailidis and Nanopoulos [42] pointed out that changes in the user preferences can vary at different rates. These changes have direct effects on the precision, novelty, and diversity of the recommendations [43, 44]. Therefore, the user preference changes need to be fully recognized to ensure the effectiveness of recommendations. Generally, the user preferences can be divided into long-term preferences and short-term preferences [18, 19, 22, 23]. The former characterizes the user general interests that are relatively stable or that change slowly over time, whereas the latter usually refers to the user temporal interests, and they can be easily influenced by a variety of factors, such as user instant demands, recent interests, and global mainstream trends in a short period of time [18].

Recently, the user long-term and short-term preferences have drawn attention from research to obtain more precise and accurate user preference profiles and improve the recommendation performance [18, 22, 23, 45]. For example, Tan and Liu [22] incorporated an attention mechanism into a recurrent neural network to capture the user preference changes and model the user long-term and short-term preferences. Similarly, Hu et al. [23] focused on user short-term preferences and developed a graph neural recommendation model that incorporates user recent activities with the attention mechanism on recurrent neural networks to explore the user short-term preferences. Furthermore, Yu et al. [18] extended the traditional recurrent neural network structure by addressing time-aware and content-aware controllers that integrate both short-term and long-term user preferences and achieve superior performance in terms of the AUC and the F1-score measures. Furthermore, Liu et al. [45] developed a hybrid attention mechanism of recurrent neural networks to capture users' long-term preferences and reinforce short-term preferences. More precisely, the authors combine the item description and visual information that makes the recommendations more apparent and interpretable and accomplish better recall and NDCG measures. Overall, recent studies have proven that distinguishing user long-term and short-term preferences could lead to more precise user profiling and achieve better recommendation performance. However, it is vital to note that the users tend to change their preferences over time. For this point, the user long-term and short-term preference profiles could have limitations if user preference changes over time are not sufficiently considered.

A number of studies (e.g., [20, 21, 26]) have explored users' preference changes over time to improve the user profiling and recommendation system performance. For example, Inuzuka et al. [21] investigated the user changes in preferences based on the user interaction with the recommendation systems, indicating that a better approach to address the user preference changes could largely improve recommendation system performance. Similarly, Rafailidis [20] focused on the pairwise correlations between the latest preferences and former preferences to better capture users' changing preferences. Lin and Chen [26] addressed user preference changes over time by separating the differentiation of the recent and the early user and item data. This study proposes a probabilistic collaborative filtering model based on the hidden Markov models (HMMs) that can obtain changes in item properties and capture the changes in the states of user preferences. According to the evidence from the previous studies, the user preference change is the critical part of forming a user profile, which must be better understood. It can be argued that without sufficient attention to an awareness of the user preference changes, the recommendation systems may still face challenges in providing the users with the recommendation outcomes that are more in line with the user personal needs. Thus, it is important to consider time awareness in the recommendation systems.

3. Research Methodology

This study will focus on providing precise, novel, and diverse top recommendations for individuals and groups of users in a business recommendation domain (e.g., restaurants, local services, hotels and entertainment facilities recommendations). To conduct the study, a time-aware hybrid recommendation approach is proposed. The proposed approach includes five components. First, neural collaborative filtering is used to select candidate items for further top personal and group recommendations. To increase the accuracy of the candidate items, users' information is exceeded by including users' gender and location, such as a city or state. Second, users' long-term preference profiles based on users' interaction history with items and items' categories are obtained. The decay function is applied for feature weight adjustment. Third, users' short-term preference profiles that reflect users' most recent preferences are obtained based on items' features extracted from users' reviews. Finally, top personal recommendations for the individual users are provided based on selected candidate items considering users' long-term and short-term preference profiles. Top group recommendations for the different groups of users are provided by aggregating the top personal recommendations.

3.1. Candidate Items Selection. Collaborative filtering is used to select accurate candidate items for personal recommendations. Specifically, neural collaborative filtering proposed by He et al. [46] has been improved by incorporating users' supplementary information, including users' gender and location, to select more precise candidate items because users with similar demographic (e.g., gender and geographic location) features tend to have similar preferences [2]. Therefore, users' gender and location are obtained in the

user gender prediction and the user location precision modules.

The user gender prediction module takes users' first names as an input and predicts the users' gender. A long short-term memory (LSTM) recurrent neural network is used in the user gender prediction module. The ability to learn long-term dependencies makes LSTM advantageous for predicting the different sequences, such as text sequences or letter sequences. The LSTM neural network consists of repeating modules. Every repeating module includes the forget layer, the input layer, and the output layer.

In the forget layer, equation (1), the sigmoid function decides what information should be discarded from the cell. The output is the value between 0 and 1 for every number in the cell:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{1}$$

where σ is the sigmoid nonlinearity, W is the weight parameter, h and x characterize the hidden output vector and the input feature vector, respectively, and b is the corresponding bias.

The input layer determines what information should be kept in the cell. It has two parts. In the first part, equation (2), a sigmoid function determines which values should be updated. In the second part, equation (3), a tanh function establishes a vector of new candidate values, $\widetilde{C}_t$:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \tag{2}$$

$$\widetilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right). \tag{3}$$

The previous cell state is updated into the new cell state. The old state is multiplied by the output from the forget layer in equation (4), and new candidate values are added:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t. \tag{4}$$

The output layer determines the output. The sigmoid function decides what segment of the cell state will be taken as the output. The function takes the cell state and gives values from −1 to 1, which decide the importance level (please see equation (5)). Finally, the tanh function is multiplied by the output of the sigmoid function in equation (6):

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right), \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right). \tag{6}$$

The user location is estimated in the user location prediction module. Specifically, the user location prediction module is developed based on interactions between the users and the items. To estimate the location of a particular user, the items that the user has interacted with are clustered by their location. The largest cluster is then selected as the user's estimated location.

For candidate item selection, neural network collaborative filtering that uses supplementary user information from the user gender prediction and the user location prediction modules is proposed (Figure 1). The proposed neural network takes items' and users' information, including gender and estimated location, as an input in the input layer. This supplementary user information is helpful to enhance the accuracy of the predicted scores. Following the input layer, the embedding layer is used to represent users' and items' information as continuous vectors. Embedding helps to indicate syntactic and semantic characteristics of items' and users' information as well as capturing the relationship among them. To convert embedded users' and items' information into one dimension, the flattened layer is applied next. Subsequently, flattened users' and items' information is concentrated together in the concentrate layer. To cope with the problem of overfitting, a dropout regularization method is used in the following dropout layer [47]. The dropout layer is followed by several hidden layers with dropout regularization between them. In the hidden layers, the model learns interactions between users' and items' latent features. These layers contain a decreasing number of neurons, which is reduced by half neurons in each hidden layer. In this way, more abstractions can be learned from the data. Finally, the output layer is represented by a single fully connected neuron that predicts the scores that users would give to unknown items. The items with predicted scores above the 3.5 threshold are selected as candidate items for further processing.

3.2. User Long-Term Preferences Profile. Users' long-term preferences are extracted from the items' description attributes (e.g., business name, business category, business features) and items' common characteristic attributes (e.g., opening hours, location, price). Although the number of description attributes and common characteristic attributes is relatively small, these attributes can precisely describe and categorize items and increase the effectiveness of building the user profiles.

To represent the users' profiles, the vector space model (VSM)-based representation was employed as a user model representation method because it takes into account the importance of the different attributes in the user profile. In the vector space model representation, a user profile P is represented as an n-dimensional vector in which each dimension matches a distinct item feature and n is the total number of these features. The example of a user profile represented in a vector space model is illustrated in

$$P_u = \{(f_1, w_1), (f_2, w_2), \ldots, (f_n, w_n)\}, \tag{7}$$

where f stands for the item feature and w indicates the importance of that feature in the user profile.

In users' long-term preferences profiles, users' features are obtained based on users' interactions with items. The initial weights of users' features are calculated by the frequency of occurrence during the interaction with items. To illustrate the process of calculating the initial weights of a particular user profile, assume that user u_1 has interacted with two items, i_1 and i_2. The first item i_1 contains the features f_1, f_2, whereas the second item i_2 contains the features f_1, f_3, and f_4. Based on the user u_1 interaction, user u_1 interacted with four features from two items. The feature f_1 is represented in both the items that the user has

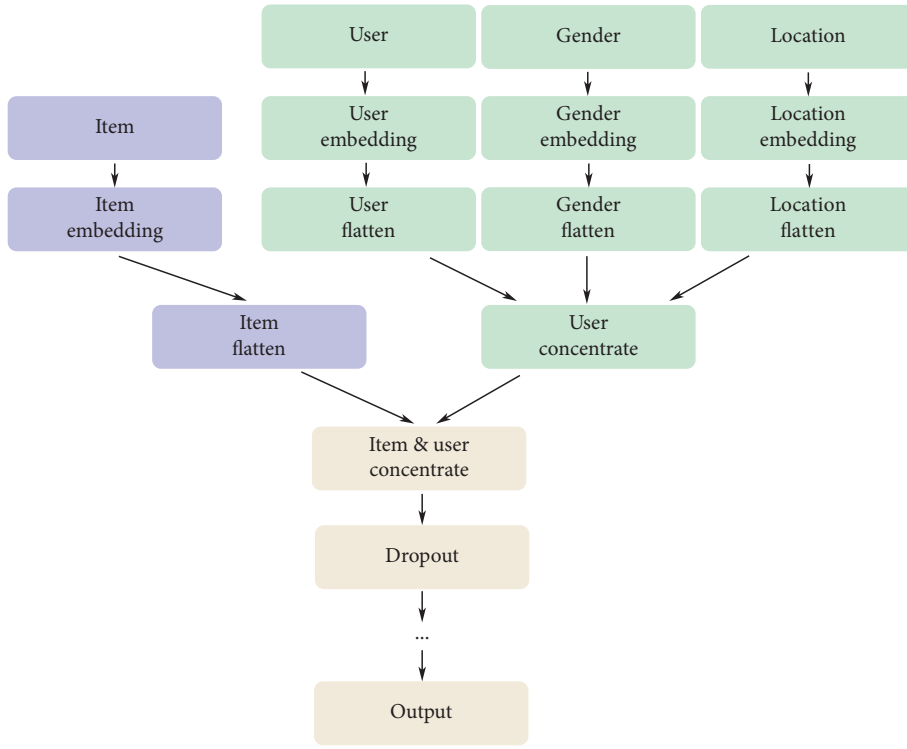


Figure 1: The architecture of the neural network.

interacted with, and its initial weight would be set as two. Whereas with features f_2, f_3, and f_4, user u_1 only interacted once; therefore, their initial weights would be set as one. However, to more accurately reflect the user preferences in the user profile, the initial feature weight requires an adjustment based on the time.

To reflect the user preference changes in a user profile, the user's feature weights decrease over time. For this purpose, the exponential decay function, equation (8), has been applied to adjust feature weights in the user's long-term preferences profile based on the time when the user has interacted with a particular item. More specifically, the items from the user's more recent interactions become more important and have a greater impact on the user's long-term preferences profile than the items from the user's earlier interactions:

$$f(t) = e^{-\alpha t * (1-\tau)}, \tag{8}$$

where α controls the decay rate, t is the time parameter, and the parameter τ determines whether particular users tends to change their preferences often or not (equation (9)). To determine the user likeness to change, the similarity between all the items that a user has interacted with is divided by the number of such items. The greater the parameter τ is, the less particular users tend to change their preferences:

$$\tau = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\text{sim}\left(I_i, I_j\right)\right)}{n}, \tag{9}$$

where n represents the number of all the items that the user interacted with.

In addition, as each user changes their preferences with different velocities, the parameter τ from equation (9) is calculated for each user, which affects the decay rate of equation (8).

3.3. User Short-Term Preferences Profile. The users' short-term preferences are obtained from the user reviews to items. The user reviews are a type of user-generated content and are represented in a free text form. All users can freely express their opinions toward items through reviews, which may be helpful to understand the reasons behind whether the users like or dislike a particular item. Thus, item features can be extracted from the users' reviews. To extract the item features, all the user reviews toward a particular item have been aggregated. These item features are extracted by using the n-gram language model algorithm [48]. The n-gram language model is a sequence of n words occurring in a given text corpus. Moreover, to extract more rich and precise features without indicating exceedingly rare ones, unigrams, bigrams, and trigrams have been considered, where n equals 1, 2 and 3, respectively. Finally, the extracted item features from the aggregated users' reviews are stored in the item profiles.

Figure 2 shows the process of obtaining the user-short-term preference profile from the user reviews. As shown in the figure, to obtain the user short-term preferences profile, the user reviews of a particular item are compared with each extracted feature through the n-gram algorithm in the item profile. To compare the user reviews with the item features, the user reviews and the item features are mapped to vectors of real numbers. To be specific, representations from the Embeddings from Language Models (ELMo) are used to represent the item features and the user

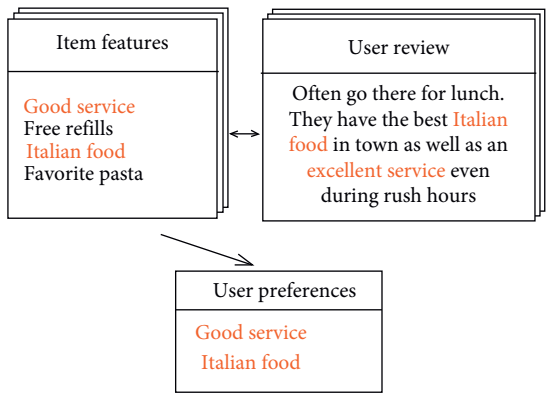


Figure 2: The process of obtaining the user short-term preferences from user reviews.

reviews in the vectors or the embeddings [49]. By doing so, syntax and semantics characteristics of items' and users' features are modelled to improve the quality of the embeddings. More importantly, contextual representation is considered by which the polysemous features can be distinguished in the embeddings from Language Models. Consequently, the similarity between item features and user reviews can be accurately calculated. After that, the most similar features are selected for the user short-term preferences profile.

In addition, a sliding-window algorithm is used to catch the most recent user preferences. The sliding-window algorithm usually considers a time-based sliding window or count-based sliding window [50]. The former relies on a user's interactions with a service from the fixed latest time interval (latest day, month, year, etc.), whereas the latter considers a fixed number of the latest user interactions with a service (latest ten, twenty, fifty, etc. interactions). Considering that each user interacts with the system differently, the sliding-window algorithm is primarily used for capturing the latest number of user interactions in the current study. Please note that based on the empirical measurements, during the process of obtaining the user short-term preferences profile, only the latest twenty user interactions are taken into consideration to reduce time complexity, improve the speed of user modeling, and catch only the recent preferences. Moreover, to obtain more precise user short-term preference changes, the user short-term preferences profile is updated dynamically every time the user interacts with a new item.

3.4. Top Personal Recommendations. The selected candidates are processed by taking the user long-term and short-term preferences profile into the top personal recommendations. The similarity between the user long-term profile attributes and the item category attributes from the selected candidates is calculated. The cosine similarity measure is used to examine the similarity between the two vectors of an inner product space. It calculates the cosine of the two vector angles:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{10}$$

where A_i and B_i are the components of vectors A and B, respectively. The output of similarity ranges from −1 to 1, where −1 indicates that two vectors are completely opposite, whereas 1 denotes that two vectors are completely similar to each other.

The item candidates that have the highest similarity with the user long-term preference profile attributes are selected and sorted in the descending order. After that, the processed items from the previous step are sorted based on a similarity score (equation (10)) between the user short-term preferences profile and the extracted item attributes from the user reviews. A final list of the top personal recommendations that considers short-term and long-term preferences is formed. The algorithm for the top personal recommendations is given as Algorithm 1 in Table 1.

As shown in Table 1, the list of the item candidates is primarily obtained for a selected user. The user short-term and long-term preference profiles are constructed. The recommendations data frame, *rec-df*, is created to store the recommendation list with the similarity scores. The similarities, *long_sim* and *short_sim*, between the item features and the user profiles are calculated and stored in the recommendations data frame. Then, the recommendations are sorted by the similarity with long-term preference profiles. The items with the greatest similarity are selected for further processing. Finally, the recommendations from the previous step are sorted by the similarity with short-term preference profiles. The final list is formed from the top items from the recommendations data.

3.5. Top Group Recommendations. The recommendations for the group of users are provided by aggregating the personal recommendations. The approach of forming the group recommendations from the personal recommendations is vital because when the recommendations are made for every user individually, it allows for catching the individual preference changes for each user in the group and dynamically reflects these changes in the final recommendations. Algorithm 2 in Table 2 is the pseudocode of the top recommendations for the groups of users.

It should be noted that the list of aggregated recommendations, *aggregatedRec*, is created first to aggregate all personal recommendations for every user in the group. The top personal recommendations for every user in the group are provided by using Algorithm 2. After that, the similarity matrix, *all_users_similarity*, between all items from aggregated personal recommendations and users in the group is created. Then, the user profiles for every user in the group are obtained. The user profiles are compared with all items from the aggregated personal recommendations, and their similarity is stored in *the aggregated similarity* list. Consequently, every user in the group and their similarity with

Table 1: The process to generate top personal recommendations.

Algorithm 1: top personal recommendations
Input: one_user**,** date, top_n
Output: List of recommended items
1 Create candidates = []
2 candidates = item candidates for one_user
3 long-term profile = Obtain long-term user preferences profile (one_user, date)
4 short-term profile = Obtain short-term user preferences (one_user, date)
5 Create rec_df data frame
6 **for** every item in candidate list **do**
7 long_sim = Calculate similarity (long-term profile, item description attributes) with equation (11)
8 short_sim = Calculate similarity (short-term profile, item attributes from reviews) with equation (11)
9 rec_df = append with long_sim, short_sim
10 **end**
11 final_rec = sort by long_sim
12 Select items with greatest long_sim in final_rec
13 final_rec = sort by short_sim
14 final_list = first top_n items from final_rec

Table 2: The process to generate top group recommendations.

Algorithm 2: top group recommendations
Input: List of users, date, top_n
Output: Final list of recommended items
1 Create aggregatedRec = []
2 **for** every user in the list of users **do**
3 personalRec = provide personal recommendations for selected user using alg. 1
4 aggregatedRec = append aggregatedRec by personalRec
5 **end**
6 Create all_users_similarity matrix
7 **for** every user in the list of users **do**
8 one_user = selected one user
9 user_profile = create user profile (one_user, date)
10 Create aggregatedSimilarity = []
11 **for** every item in aggregatedRec **do**
12 similarity = similarity between user_profile and selected item from aggregatedRec
13 aggregatedSimilarity = append aggregatedRec by similarity
14 **end**
15 all_users_similarity = insert selected user and aggregatedSimilarity
16 **end**
17 Calculate the average similarity in all_users_similarity
18 Sort values by average similarity in all_users_similarity
19 final = first top_n items from all_users_similarity

every item are added to the similarity matrix *all_users_similarity*. The average similarity between the users and every item in the similarity matrix is then calculated. Finally, the items are sorted by the average similarity in the descending order, and the final top items are selected for the recommendations to the group of users. It should be mentioned that Algorithm 2 can provide top recommendations for any number of users in the group.

4. Experimental Study

In this section, an experimental study is conducted to evaluate the performance of our proposed time-aware hybrid approach for individual and group recommendations. This proposed approach is measured by providing the top 5, 10, and 20 recommendations for individuals as well as groups of users in the business sector (e.g., restaurants, local services, hotels and entertainment facilities recommendations).

4.1. Datasets. The Yelp dataset is used in this study. The dataset contains business data and user data [51]. Business data include precise business information, such as city, state, latitude and longitude, stars, review count, business attributes, business open hours, and categories. User data contain rich information, such as user friend mapping, user name, user ID, number of reviews, average given stars, review data, and registration date (please see the examples in Tables 3 and 4). Initially, the dataset contained 6,685,900 reviews of 192,609 businesses. However, to better capture the users' preference changes at different time points, the dataset was further refined and data were reselected, retaining the

Table 3: Review data.

Review	User	Business	Stars	Date	Text_review
345	633	43	5	2016-05-28	Lester's is located . . .
124	752	321	4	2018-03-15	Love coming here . . .
5322	23	1873	4	2017-02-21	Had their chocolate. . .

Table 4: User data.

User	Name	Review count	Registration date	Average stars
5	Kenny	531	2011-08-10	3.75
342	Eric	426	2014-01-15	4.16
834	Stan	391	2015-04-29	4.26

most active users and yielding a final 313,261 user reviews of 78,138 businesses. This dataset is split into two parts: a training set that contains 70% of the data and a test set that covers 30% of the data.

In addition, to train the LSTM recurrent neural network in the user gender prediction module, the National Data that depict the frequency of individuals' given names in the United States with an associated social security number are used in this study [52]. The data are based on social security records on March 3, 2019, and contain the records of given names for more than a hundred years. In total, the data include 98,399 unique names. The data cover the fields of name, gender, and the frequency of a name. This dataset is divided into the training, testing, and validation sets according to proportions of 60%, 20%, and 20%, respectively.

4.2. The User Changes Preferences Example. The users change their preferences at different rates. Here, an example is provided to illustrate how user preferences changes over a long period of time based on the user long-term preference profiles (from 2013 to 2019). Two users and one item were selected randomly from the Yelp dataset. Changes in the users' preferences can be clearly identified by measuring the similarity between the users' preferences and an item's features throughout a period of time. The similarity values with high discrepancy indicate that the users tend to change their preferences more frequently and dramatically. The similarity between users' long-term preference profiles and the selected item is measured by equation (11) throughout the selected period. Figure 3 shows the preference changes of two users toward the same item. It can be seen that a significant difference in preferences from 2013 to 2019 is found for both users. Moreover, the two users tend to change their preferences at different rates. More specifically, it seems that one user has more stable preferences over a certain period, whereas the preferences of another user change more frequently and dramatically. These findings further confirm that the rate of users' preference changes differs, and it should be addressed in the recommendation process for the individuals and the groups of users.

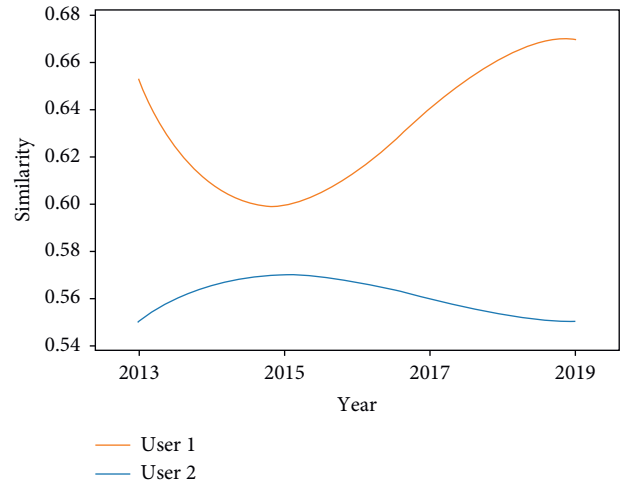


Figure 3: User changing similarity with an item throughout time.

4.3. Evaluation Metrics. To evaluate the performance of the proposed recommendation approach, a variety of evaluation metrics, including precision, recall, novelty, and diversity metrics, are used in the study. By focusing on these metrics, it can comparatively highlight the efficiency of the evaluation. All evaluation metrics are applied at a top k rank, where only the top k results in the recommendation list are considered. Precision and recall evaluation metrics are represented in equations (11) and (12), respectively:

$$P@k = \frac{1}{|U|}\sum_{u\in U} P_u@k, P_u@k = \frac{T_u@k}{k}, \tag{11}$$

$$R@k = \frac{1}{|U|}\sum_{u\in U} R_u@k, R_u@k = \frac{T_u@k}{N_u}, \tag{12}$$

where k is the length of the recommendation list; in the experimental studies, k corresponds to 5, 10, and 20. U is the set of users in the set, $P_u@k$ is the precision at k for a given user u, $R_u@k$ is the recall at k for a given user u, $T_u@k$ is the number of the user's target items that the user likes in the recommendation list, and N_u is the total number of the user's target items in the test set.

The novelty metric defines how unfamiliar and surprising recommended items are to a particular user [13] (see equation (13)). To measure the item's novelty, its probability is defined as a function of the item's rank for all the users. Therefore, the novelty for all the items in the recommendation list is defined as the average popularity rank:

$$N = \sum_{i\in R}\frac{\log_2 P_i}{n}, \tag{13}$$

where R is the list of top recommendations provided for a given user, and n stands for the number of provided recommendations.

The diversity metric stands for the average dissimilarity score between every possible pair of recommended items for a particular user (please refer to equation (14)). It denotes the level of difference between the recommended items:

$$D = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(1 - S_{i,j}\right)}{(n/2) * (n-1)}, \tag{14}$$

where S is the similarity score between every possible pair of recommended items, and n represents the number of recommendations for a given user. The similarity score between two items is calculated by using the cosine similarity measure (as presented in equation (10)). A higher similarity score among recommended items indicates a low level of diversity.

4.4. Personal Recommendations Evaluation Design. To examine the effectiveness of our proposed approach to user preference changes, top personal recommendations are provided at two time points (T1 and T2). As long-term preferences tend to change slowly over time, the effects of the user preference changes could be more obvious over a long time interval. On that premise, the two-year time interval was chosen between T1 and T2, which represent the date of 11-11-2017 (T1) and the date of 11-11-2019 (T2), respectively, where T2 is the date on which the updated dataset was obtained. The recommendation performance is evaluated by precision, recall, novelty, and diversity. Moreover, the results of the top personal recommendations are compared with a set of baseline algorithms, including the K-nearest neighbors algorithm (K-NN) [53], K-means clustering algorithm [54], co-clustering [55], nonnegative matrix factorization (NMF) [56], and singular value decomposition (SVD) algorithm [57].

4.5. Group Recommendations Evaluation Design. To conduct the evaluation of our group recommendation approach, recommendations are provided for groups with different numbers of users, including small, medium, and large groups. Small, medium, and large groups contain 3, 6, and 12 users, respectively. For every group size, 100 groups are randomly generated from the dataset. In total, 300 groups containing 2100 users are used for group recommendation evaluation.

The proposed group recommendation approach is compared with different group recommendation approaches, including the average, the least misery, and the most pleasure approaches based on the neural collaborative filtering algorithm [46]. The neural collaborative filtering average approach (average-nnmf) considers the opinion of every user in the group equally and takes into account the average of predicted ratings for each user in the group for a particular item and employs it for the group predictions for that item. The neural collaborative filtering least misery approach (lm-nnmf) attempts to minimize the misery for the users in the group. The main idea is that the group is as satisfied with the predictions as the least satisfied user in the group. In this approach, the predictions for a group correspond to the minimum of the predicted rating of each user in the group for a particular item. Finally, neural collaborative filtering of the most pleasure approach (mp-nnmf) takes into consideration the items that one user in the group likes the most but does not take into account other user preferences in the group. The predictions of a group correspond to the maximum of the predicted rating of each user in the group for a particular item.

5. Results and Discussion

5.1. Results of Top Personal Recommendations. Figure 4 depicts the results of the top 5, 10, and 20 personal recommendations in terms of precision. Overall, the proposed approach has higher scores at both T1 and T2, which shows that our proposed approach has better performance in precision than the other algorithms. It implies that the proposed approach could identify and provide more relevant items for users. Furthermore, it can be seen that there is a difference in recommendation performance between T1 and T2. More specifically, the performance of the proposed approach at T1 reaches 0.8683, 0.8492, and 0.8331 at the top 5, 10, and 20 recommendations, respectively. Our proposed approach at T2 shows a better performance, reaching 0.8723, 0.8521, and 0.8340 for the top 5, 10 and, 20 recommendations, respectively. This difference may indicate that preferences in users' profiles were constantly changing and, as a result, had different weights at T1 and T2. These changes at both the time points have been captured by our proposed approach. This implies that changes in the user preferences can moderately affect the recommendation performance, especially in precision. In addition, it is interesting to see that the performance of the proposed approach gradually declines from the top 5 to 20 recommendations throughout T1 and T2. A possible explanation may be that the increasing number of recommendations may have more irrelevant (or false positive) items, which in turn decreases the recommendation precision.

Moreover, the results of the recommendation performance regarding the recall are shown in Figure 5. As presented in the figure, among the measure approaches, the performance of our proposed approach at both the time points is placed on the top, showing a better recommendation performance. More specifically, the proposed approach has a higher proportion of relevant items in top 5, 10, and 20 recommendations. However, it is worth noting that the performance at T1 is very close to the performance at T2. A possible explanation is that the percentage of relevant (true-positive) candidate items remains stable at different time points. This may have occurred because the changes in the user preferences have small effects on recall performance. Surprisingly, the results indicate that the performance at T1 is better than the performance at T2. Such results are not consistent with the findings of the precision, showing that the recommendation at T2 has a better performance than T1. This may be attributed to a consequence of the precision-recall tradeoff, where the increases of one metric (precision or recall) can lead to the decreases of another, and vice versa. Our results also show that the performance in terms of the recall increases gradually from the top 5 to top 20 recommendations at both T1 and T2. This may suggest that a positive correlation exists between the proportion of relevant (or true positive) items and the number of recommendations.

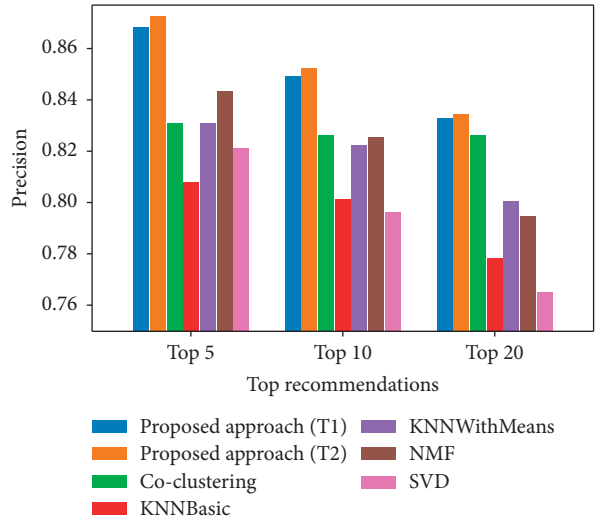


Figure 4: Precision at (k) where $k = 5$, 10, and 20.

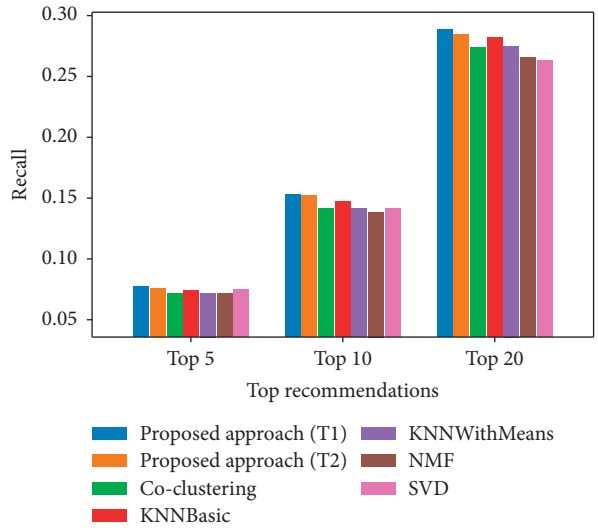


Figure 5: Recall at (k) where $k = 5$, 10 and 20.

Regarding the results of the recommendation performance in terms of the novelty (see Figure 6), overall, the results show that the proposed approach has a better performance than the other algorithms. However, it is worth noting that the performance of the proposed approach at T2 slightly underperforms the K-NN and co-clustering algorithms at the top 10 recommendations. This may be indicative of the user preferences that have stabilized at T2, which leads to a lower level of performance regarding novelty. Moreover, performance differences in aspects of novelty are found between T1 and T2. Such differences are reflected by the changes of user profiles and features' weights, showing that the performance of novelty is greatly influenced by changes in user preferences.

Figure 7 presents the recommendation performance of the diversity at the top 5, 10, and 20 personal recommendations. As expected, the results show that the proposed approach has better performance in terms of the diversity among all measure approaches. It implies that the items recommended by the proposed approach are more diverse and dissimilar. Furthermore, it is notable that there are significant differences in the recommendation performance between T1 and T2. Specifically, the diversity of the proposed approach achieves 1.8038, 1.7823, and 1.7710 at the top 5, 10 ,and 20 recommendations at T1, respectively. The performance at T2 reaches 1.7938 at the top 5, 1.7743 at the top 10, and 1.755 at the top 20 recommendations, which is lower than the performance at T1. It can be explained that the recommendation performance of the diversity might be influenced by the users with the less preference changes at T2. Such performance differences further support our previous findings that the weights of the preferences in users' profiles are distinct at T1 and T2. Moreover, such results show that the diversity metric can be more affected by user preference changes. In addition, it is interesting to note that the performance of the diversity decreases from the top 5 to

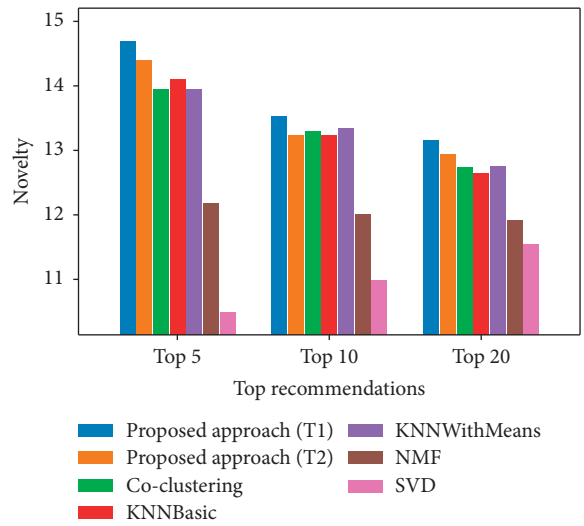


Figure 6: Novelty at (k) where $k = 5$, 10, and 20.

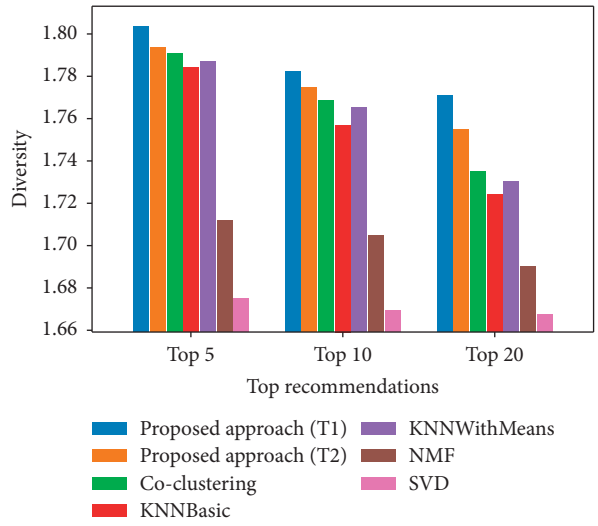


Figure 7: Diversity at (k) where $k = 5$, 10, and 20.

the top 20 recommendations at both T1 and T2. Such a finding can be explained by considering that the more the items are recommended, the fewer the dissimilarities that can be found.

Furthermore, the average performance of each measurement approach in terms of the precision, recall, novelty, and diversity metrics is shown in Table 5. The results show that the scores for the proposed approach are higher than other approaches in each metric, indicating that the proposed approach has a better average performance (highlighted in bold in the table). Such results confirm that better performance in terms of precision, novelty, and diversity can increase overall performance. In addition, the average performance between T1 and T2 shows the differences in terms of precision, novelty, and diversity. This implies that the recommendation performance is affected by the time differently, which in turn notes the importance of users' preference changes. These findings suggest that capturing users' preference changes can not only make precision improvements but also enhance the novelty and the diversity of the recommendations. At this point, our proposed approach is useful for efficiently providing recommendations to the users who either continuously change their preferences or keep stable preferences.

5.2. Results of the Top Group Recommendations. The results of the group recommendations in terms of the precision among the small, medium, and large groups are presented in Figure 8. Overall, the results show that the proposed approach outperforms other approaches within all three types of group recommendations. It shows that more relevant items are recommended by the proposed approach than other approaches within the small, medium, and large groups of users. Regarding the performance of the proposed approach, the results show that a better performance is

Table 5: Average precision, recall, novelty, and diversity.

	Precision	Recall	Novelty	Diversity
Co-clustering	0.8276	0.1622	13.3255	1.7647
KNN basic	0.7959	0.1676	13.3268	1.7552
KNN with means	0.8181	0.1626	13.3496	1.7607
NMF	0.8212	0.1581	12.0314	1.7024
SVD	0.7939	0.1597	11.0111	1.6705
Proposed approach (T1)	**0.8501**	**0.1729**	**13.7935**	**1.7877**
Proposed approach (T2)	**0.8522**	**0.1709**	**13.5268**	**1.7744**

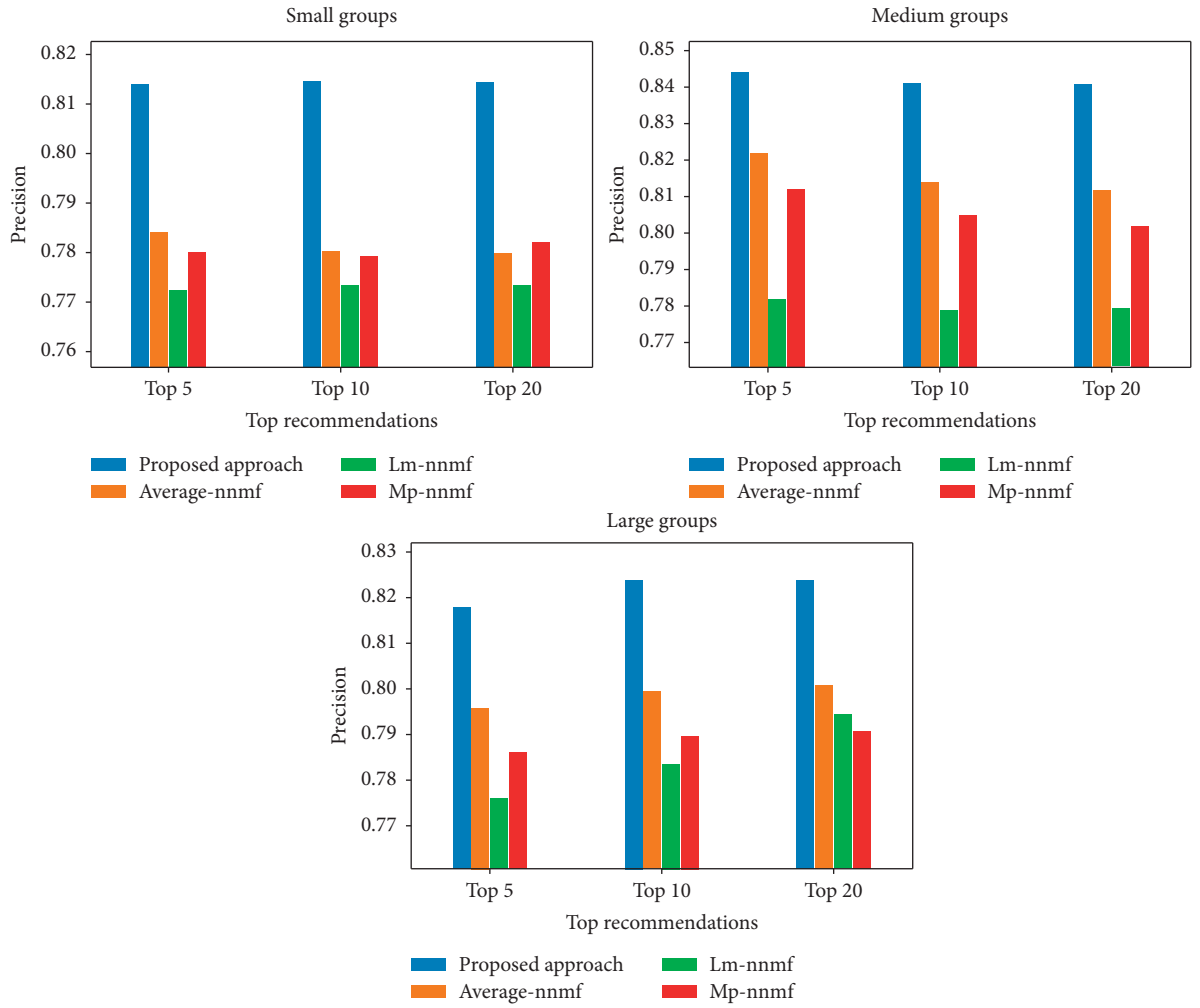


Figure 8: Precision of the three types of group recommendations' evaluation.

achieved by the medium group. The large group places next, and the small group performs worse. This may have occurred because the distribution of the preferences among the users of the medium groups is relatively low, whereas users in the small and the large groups have a high level of preference distribution. Such a low distribution in medium groups indicates that the users have similar preferences, and the proposed approach could better identify relevant items for groups and consequently achieve better precision performance. Moreover, it shows that the number of users in the groups has an impact on the precision performance within the three groups. This increasing number of users in a group may challenge the precision of recommendation. In addition, it is interesting to note that the precision of the top group recommendations remains stable from the top 5 to the top 20 recommendations among all the three types of groups. This indicates that the increasing number of recommendations does not affect the ratio of irrelevant items, which leads to stable performance of the precision.

Figure 9 depicts the recommendations performance in terms of the recall among the three groups of users. As shown in the figure, overall, the proposed approach achieves better recommendation performance than other approaches among the three groups. It seems that the proposed

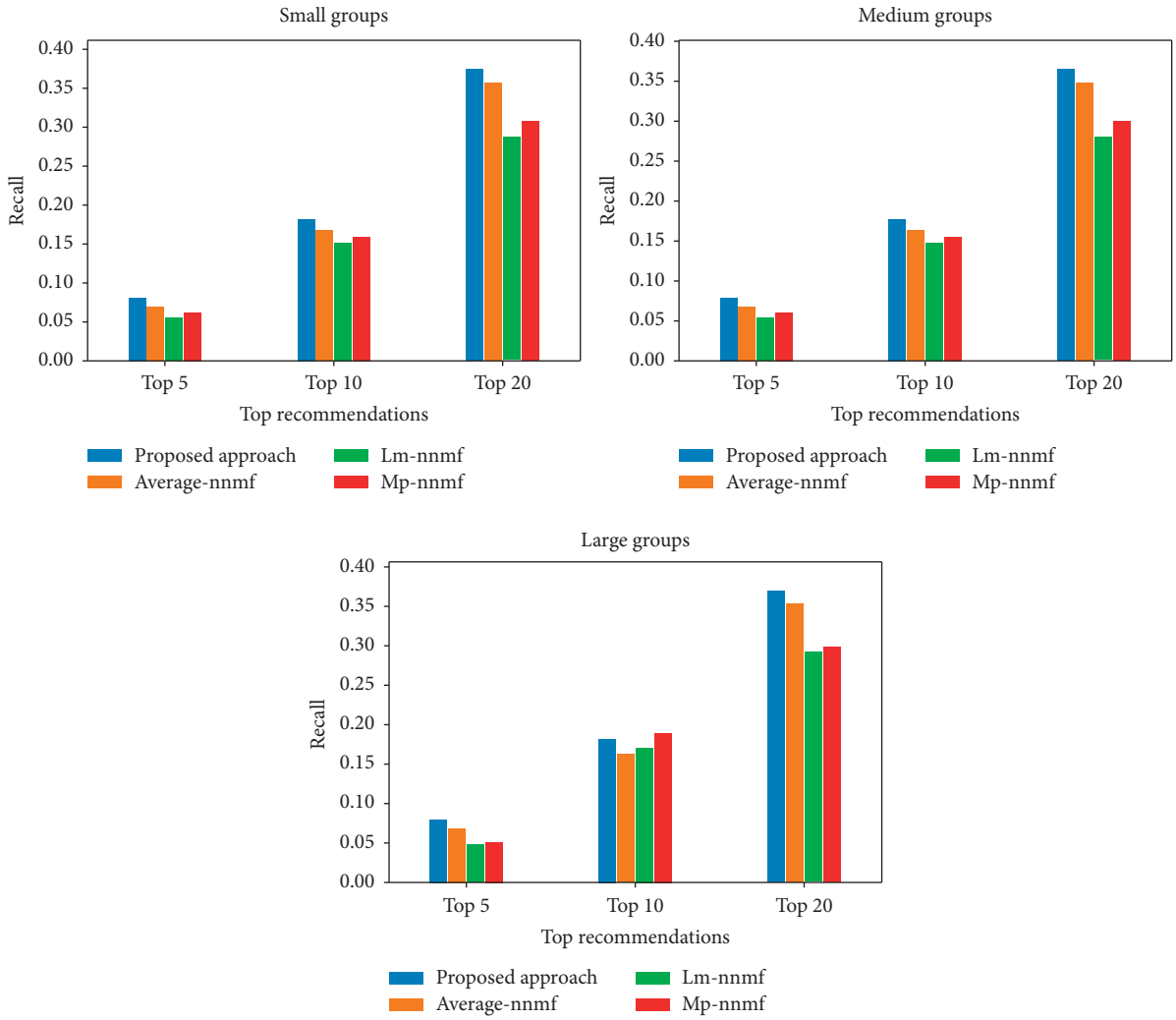


Figure 9: Recall of three types of group recommendations' evaluation.

approach provides the higher percentage of relevant (true-positive) items among all the groups. However, it is interesting to see that in large groups, the proposed approach slightly underperforms the *mp-nnmf* approach at the top 10 recommendations. This may have occurred because the proposed approach inaccurately identified irrelevant items for the large groups of users at the top 10 recommendations due to a high level of preference distribution among users in the large group. Consequently, the rate of relevant (true-positive) items decreased, leading to the lower performance of the recall. Moreover, it can be noted that in the proposed approach, the results of the recall within the three groups are fairly close to each other. Such results indicate that the users in the different groups may have small effects on the recommendation performance of the recall. In addition, the results show that the performance of the recall within each group has a significant increase from the top 5 to the top 20 recommendations. This may imply that the increasing number of recommendations may have more relevant (or true positive) items, which increases the recommendation recall.

Figure 10 shows the performance results of the novelty at the top 5, 10, and 20 recommendations among the small, medium, and large groups. As expected, it seems clear that the proposed approach outperforms other approaches among all groups. Moreover, the results reveal that the novelty in small and medium group recommendations follows the same pattern in which the performance decreases gradually from the top 5 to the top 20 recommendations. Such a decrease can possibly be explained by the fact that the more items are recommended, the more high-ranked items are included in recommendations. Nonetheless, for the large groups, the novelty decreases steadily from the top 5 to the top 10 recommendations with a sudden increase from the top 10 to the top 20 recommendations. A possible explanation of such a fluctuation lies in the distribution of preferences among the users in the groups. This supports our previous findings and proves that large groups contain users with widely differing preferences that affect recommendation outcomes in novelty. Furthermore, the values of the results show a high discrepancy among small, medium, and large groups. This discrepancy can be caused by the size of

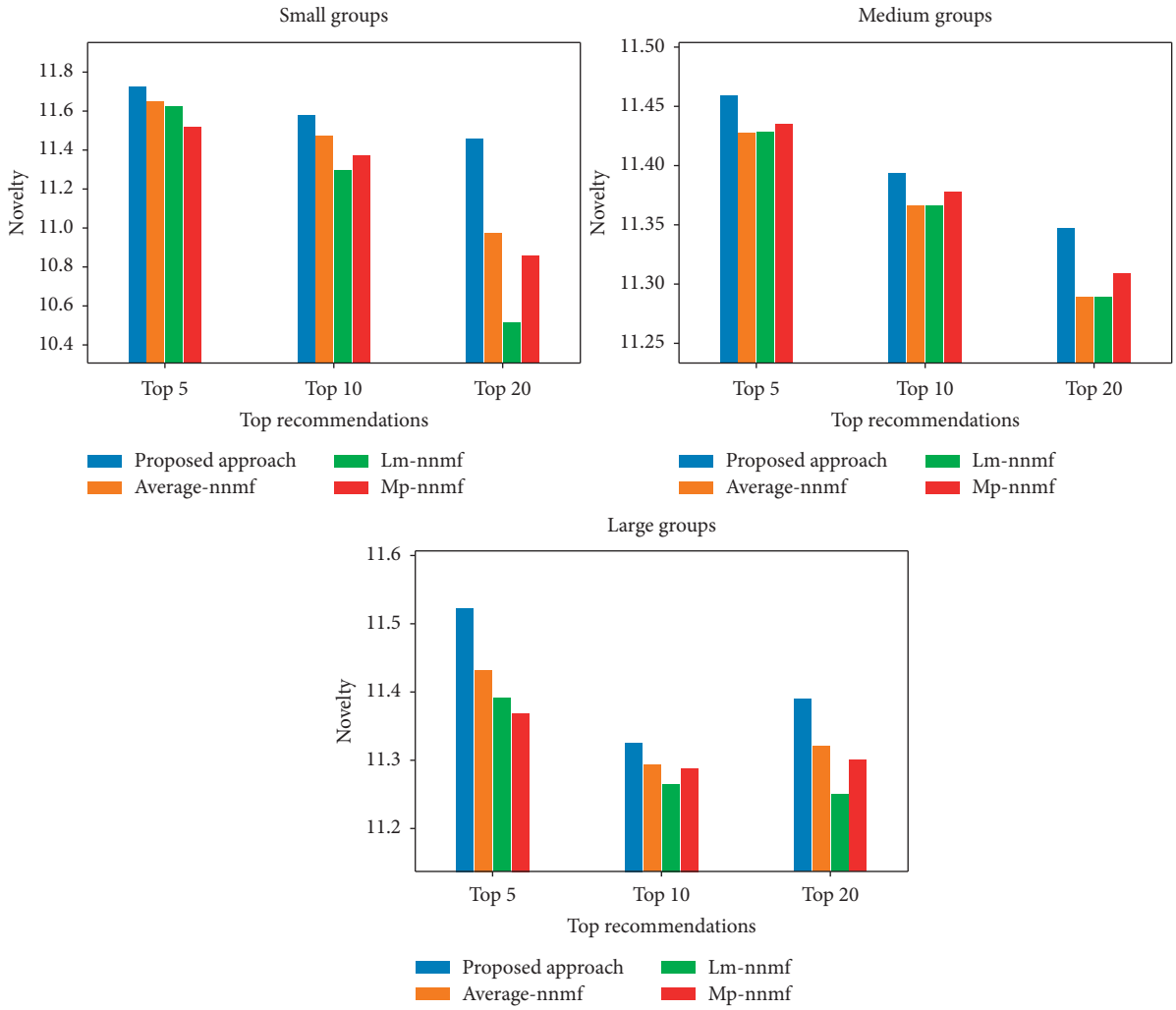


Figure 10: Novelty of the three types of group recommendations evaluation.

the user groups. Thus, the number users in a group could significantly affect the performance of the novelty.

Furthermore, the performance results of the diversity at the top 5, 10, and 20 recommendations among the three groups are shown in Figure 11. It is clear that the proposed approach shows better performance than other approaches in small, medium, and large groups. In particular, the diversity of recommendations for the small groups reaches 1.095, 1.232, and 1.382 for the top 5, 10, and 20 recommendations, respectively. In the medium groups, it achieves 0.979 for the top 5, 1.086 for the top 10, and 1.204 for the top 20 recommendations. A similar increasing trend can be seen in the large group recommendations, which hit 0.931, 1.025, and 1.112 for the top 5, 10, and 20 recommendations, respectively. Moreover, the results show that the performance of the diversity increases from the top 5 to the top 20 recommendations among all the three types of groups. This can possibly be explained by the fact that the more items are recommended, the more dissimilarities are found among recommended items. However, a negative correlation between the number of users in the group and diversity performance is found in the results. It may be explained by the distribution of users' preferences in the different groups. As there is an increase of users in the groups, more users may share interchangeable preferences. Accordingly, the similarity among recommended items rises and consequently influences the performance of diversity.

Finally, the average performance of precision, recall, novelty, and diversity in the small, medium, and large groups is presented in Table 6. The results are consistent with our previous findings showing that the performance of the proposed approach is better than other approaches in all evaluation metrics among the different sizes of group recommendations. Moreover, the results show that there are differences in performance among small, medium, and large groups. Specifically, the performance of recall shows small differences of results among small, medium, and large groups, which implies that users in groups only have a slight impact on the recommendation performance in recall. However, the differences in the results in precision, novelty, and diversity among all the three groups are relatively high, which implies that the users have a considerable impact on the recommendation performance. These findings show that considering the users in groups could enhance the overall

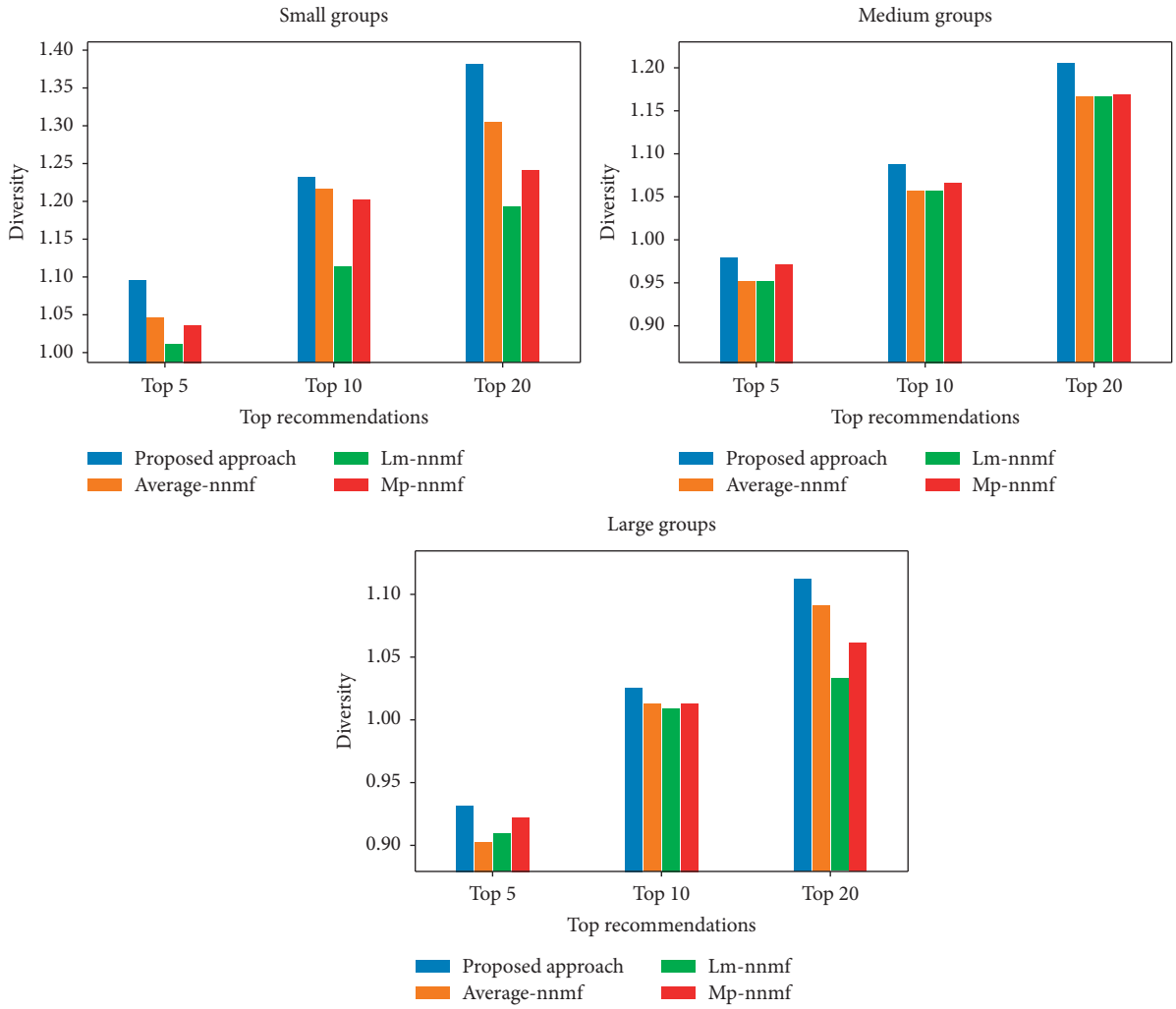


Figure 11: Diversity of the three types of group recommendations' evaluation.

Table 6: Average precision, recall, novelty, and diversity.

		Precision	Recall	Novelty	Diversity
Small groups	Average-nnmf	0.7813	0.1982	11.3706	1.1895
	Lm-nnmf	0.7731	0.1642	11.1478	1.1066
	Mp-nnmf	0.7804	0.1755	11.2481	1.1604
	Proposed approach	**0.8152**	**0.2127**	**11.5811**	**1.2367**
Medium groups	Average-nnmf	0.8161	0.1892	11.3686	1.0529
	Lm-nnmf	0.7801	0.1555	11.3621	1.0576
	Mp-nnmf	0.8063	0.1599	11.3774	1.0686
	Proposed approach	**0.8421**	**0.2051**	**11.3997**	**1.0904**
Large groups	Average-nnmf	0.7988	0.1949	11.3481	1.0029
	Lm-nnmf	0.7846	0.1707	11.3013	0.9852
	Mp-nnmf	0.7888	0.1818	11.3185	0.9990
	Proposed approach	**0.8221**	**0.2128**	**11.4121**	**1.0231**

recommendation performance and provide not only precise but also novel and diverse group recommendations.

6. Conclusion

The recommendation systems work as a filtering service to fight against information overload, which is increasingly drawing the attention of the practitioners and the academics. Evidence from previous studies indicates that there is a need to develop an efficient recommendation approach to offer precise positioning of information targets and facilitate resource utilization not only for individuals but also for the groups of users. In particular, there is a need to address time awareness, which can be more in line with user preference

changes to enable the improvement of recommendation performance. Therefore, this study aims to address user preference changes through a certain period of time to develop a hybrid recommendation approach that provides recommendations for both the individuals and the groups of users. More specifically, the following research questions are investigated: (1) What are the effects of the user preferences changes on the individual user and group recommendations? and (2) How do user groups affect the precision, recall, novelty, and diversity of group recommendations? To answer these questions, our proposed approach integrates neural collaborative filtering with a content-based method for individual recommendations, and the personal recommendation aggregation method is employed for group recommendations.

The results are summarized in the following aspects. First, the overall results show that the proposed approach achieves better performance than other algorithms, including K-NN, K-means, co-clustering, NMF, and SVD in personal recommendations and average-nnmf average-nnmf, mp-nnmf in group recommendations. Such results further confirm the validation of our proposed approach for individual and group recommendations. Second, the differences in the recommendation performance in aspects of precision, recall, novelty, and diversity between T1 and T2 are found in the study. This may imply that users' dynamic preference changes through time can be well captured in users' profiles by our proposed approach. Moreover, the results indicate that the changes in user preferences over time have a great impact on recommendation performance. These findings provide evidence that addressing user preference changes over time improves user modeling and consequently increases recommendation performance. Third, the results of the group recommendation evaluation demonstrate that the performance of precision, recall, novelty, and diversity differs among the small, medium, and large groups of users. These findings reveal that the number of users in groups has a considerable impact on the efficiency and effectiveness of group recommendation performance.

This study contributes to recommendation systems research by developing a time-aware hybrid recommendation approach to offer recommendations for both the individual and the group users. Furthermore, investigating the link between the user changing preferences over time and the user profiles provides deep insights into building more effective and precise user profiles that could improve the recommendation performance. The proposed approach would be helpful for developers to increase their understanding of capturing the user dynamic preference changes in the user profiles as well as strengthening the knowledge of the user profile modeling for both the personal and the group recommendations. In addition, our study can also help scholars broaden their knowledge of recommendation evaluation by focusing on multiple metrics, including precision, recall, novelty, and diversity, thereby providing deeper insights into measuring the efficiency of the recommendation systems.

This study has some limitations. For example, the proposed approach only considers users' positive feedback in the user modeling without addressing the negative feedback, which may limit the modeling user profiles and the recommendation performance. Further study can address both positive and negative feedback to obtain more comprehensive users' preferences to improve the user profiles. Furthermore, the proposed approach is limited by neglecting the social relationships among the users in the group, which may affect group satisfaction and the quality of group recommendations [58]. Future work can extend the study of the relations among the users in the group, including friends, colleagues, relatives, or strangers. The results will be beneficial for optimizing and improving the group recommendations.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by research grants funded by the National Natural Science Foundation of China (grant no. 61771297).

References

[1] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: issues, challenges, and research opportunities," in *Lecture Notes in Electrical Engineering*, pp. 1179–1189, Springer, Singapore, 2016.

[2] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Springer, New York, NY, USA, 2015.

[3] V. Subramaniyaswamy and R. Logesh, "Adaptive KNN based recommender system through mining of user preferences," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2229–2247, 2017.

[4] I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: a systematic review," *Expert Systems with Applications*, vol. 97, pp. 205–227, 2018.

[5] A. Gunawardana and S. Guy, "A survey of accuracy evaluation metrics of recommender tasks," *Journal of Machine Learning Reearch*, vol. 10, pp. 2935–2962, 2009.

[6] F. Zafari, I. Moser, and T. Baarslag, "Modelling and analysis of temporal preference drifts using a component-based factorised latent approach," *Expert Systems with Applications*, vol. 116, pp. 186–208, 2019.

[7] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? a survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2017.

[8] K. Tarnowska, Z. W. Ras, and L. Daniel, "Studies in big data recommender system for improving customer loyalty," *Conclusions*, vol. 55, pp. 123-124, 2019.

[9] A. Gogna and A. Majumdar, "Balancing accuracy and diversity in recommendations using matrix completion

framework," *Knowledge-Based Systems*, vol. 125, pp. 83–95, 2017.

[10] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage," *ACM Transactions on Interactive Intelligent Systems*, vol. 7, no. 1, pp. 1–42, 2017.

[11] T. Yu, J. Guo, W. Li, H. J. Wang, and L Fan, "Recommendation with diversity: An adaptive trust-aware model," *Decision Support Systems*, vol. 123, Article ID 113073, 2019.

[12] D. M. Fleder and K. Hosanagar, "Recommender systems and their impact on sales diversity," in *Proceedings of the 8th ACM Conference on Electronic Commerce*, San Diego, CA, USA, 2007.

[13] N. Hurley and M. Zhang, "Novelty and Diversity in top-N recommendation analysis and evaluation," *ACM Transactions on Internet Technology*, vol. 10, no. 4, pp. 1–30, 2011.

[14] M. Kompan and M. Bielikova, "Group recommendations: survey and perspectives," *Comput. Informatics*, vol. 33, no. 2, pp. 446–476, 2014.

[15] G. Rebala, A. Ravi, and S. Churiwala, *Recommender Systems. An Introduction to Machine Learning, Springer International Publishing, New York City, NY, USA*, pp. 169–179, 2019.

[16] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, 2012.

[17] T. D. Pessemier, S. Dooms, and L. Martens, "Comparison of group recommendation algorithms," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 2497–2541, 2013.

[18] Z. Yu, J. Lian, A. Mahmoody, G. Liu, and X. Xie, "Adaptive user modeling with long and short-term preferences for personalized recommendation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China, 2019.

[19] R. Devooght and H. Bersini, "Long and short-term recommendations with recurrent neural networks," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, Bratislava, Slovakia, 2017.

[20] D. Rafailidis, "A Multi-Latent Transition model for evolving preferences in recommender systems," *Expert Systems with Applications*, vol. 104, pp. 97–106, 2018.

[21] K. Inuzuka, T. Hayashi, and T. Takagi, "Recommendation system based on prediction of user preference changes," in *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence*, Omaha, NE, USA, 2016.

[22] Q. Tan and F. A. Liu, "Recommendation based on users' long-term and short-term interests with attention," *Mathematical Problems in Engineering*, vol. 2019, Article ID 7586589, 13 pages, 2019.

[23] L. Hu, C. Li, C. Shi, C. Yang, and C. Shao, "Graph neural news recommendation with long-term and short-term interest modeling," *Information Processing & Management*, vol. 57, no. 2, Article ID 102142, 2020.

[24] F. Eskandanian and B. Mobasher, "Detecting changes in user preferences using hidden markov models for sequential recommendation tasks," 2018, http://arxiv.org/abs/1810.00272.

[25] D. Rafailidis, P. Kefalas, and Y. Manolopoulos, "Preference dynamics with multimodal user-item interactions in social media recommendation," *Expert Systems with Applications*, vol. 74, pp. 11–18, 2017.

[26] Z. Lin and H. Chen, "Recommendation over time: a probabilistic model of time-aware recommender systems," *Science China Information Sciences*, vol. 62, no. 11, 2019.

[27] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[28] G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma, and X. Wang, "Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems," *Knowledge-Based Systems*, vol. 138, pp. 202–207, 2017.

[29] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 239–249, 2019.

[30] Y. Wang, M. Wang, and W. Xu, "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8263704, 9 pages, 2018.

[31] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, "EARS: emotion-aware recommender system based on hybrid information fusion," *Information Fusion*, vol. 46, pp. 141–146, 2019.

[32] J. K. Tarus, Z. Niu, and D. Kalui, "A hybrid recommender system for e-learning based on context awareness and sequential pattern mining," *Soft Computing*, vol. 22, no. 8, pp. 2449–2461, 2017.

[33] N. R. Kermany and S. H. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electronic Commerce Research and Applications*, vol. 21, pp. 50–64, 2017.

[34] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Information Sciences*, vol. 417, pp. 72–87, 2017.

[35] Y.-D. Seo, Y.-G. Kim, E. Lee, K.-S. Seol, and D.-K. Baik, "An enhanced aggregation method considering deviations for a group recommendation," *Expert Systems with Applications*, vol. 93, pp. 299–312, 2018.

[36] Z. Guo, C. Tang, H. Tang, Y. Fu, and W. Niu, "A novel group recommendation mechanism from the perspective of preference distribution," *IEEE Access*, vol. 6, pp. 5865–5878, 2018.

[37] Z. Guo, W. Zeng, H. Wang, and Y. Shen, "An enhanced group recommender system by exploiting preference relation," *IEEE Access*, vol. 7, pp. 24852–24864, 2019.

[38] N. Capuano, F. Chiclana, E. Herrera-Viedma, H. Fujita, and V. Loia, "Fuzzy group decision making for influence-aware recommendations," *Computers in Human Behavior*, vol. 101, pp. 371–379, 2019.

[39] C. Villavicencio, S. Schiaffino, J. A. Diaz-Pace, and A. Monteserin, "Group recommender systems: a multi-agent solution," *Knowledge-Based Systems*, vol. 164, pp. 436–458, 2019.

[40] O. Kaššák, M. Kompan, and M. Bieliková, "Personalized hybrid recommendation for group of users: Top-N multimedia recommender," *Information Processing & Management*, vol. 52, no. 3, pp. 459–477, 2016.

[41] T. D. Pessemier, J. Dhondt, and L. Martens, "Hybrid group recommendations for a travel service," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2787–2811, 2016.

[42] D. Rafailidis and A. Nanopoulos, "Modeling the dynamics of user preferences in coupled tensor factorization," in *Proceedings of the 8th ACM Conference on Recommender Systems*, Foster, CA, USA, 2014.

[43] D. Rafailidis and A. Nanopoulos, "Modeling users preference dynamics and side information in recommender systems,"

IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46, no. 6, pp. 782–792, 2016.

[44] C. Rana and S. K. Jain, "A study of the dynamic features of recommender systems," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 141–153, 2012.

[45] P. Liu, L. Zhang, and J. A. Gulla, "Dynamic attention-based explainable recommendation with textual and visual fusion," *Information Processing & Management*, vol. 57, Article ID 102099, 2019.

[46] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[48] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, "Word n-gram attention models for sentence similarity and inference," *Expert Systems with Applications*, vol. 132, pp. 1–11, 2019.

[49] M. Peters, M. Neumann, and M. Iyyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, LA, USA, 2018.

[50] X. Wang, Y. Zhang, and W. Zhang, "Skype," in *Proceedings of the VLDB Endowment*, vol. 9, pp. 588–599, 2016.

[51] Y. Dataset, https://www.yelp.com/dataset, 2020.

[52] National data of popular names. (2020). https://www.ssa.gov/oact/babynames/names.zip.

[53] D. Chae, S. Lee, S. Lee, and S. Kim, "On identifying k -nearest neighbors in neighborhood models for efficient and effective collaborative filtering," *Neurocomputing*, vol. 278, pp. 134–143, 2018.

[54] U. Kużelewska, "Collaborative filtering recommender systems based on k-means multi-clustering," *Contemporary Complex Systems*, vol. 761, pp. 316–325, 2018.

[55] K. Honda, "Fuzzy Co-clustering and application to collaborative filtering," in *Proceedings of the International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, pp. 16–23, Da Nang, Vietnam, 2016.

[56] S. Yang and L. Zhang, "Non-redundant multiple clustering by nonnegative matrix factorization," *Machine Learning*, vol. 106, no. 5, pp. 695–712, 2016.

[57] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data," *Knowledge-Based Systems*, vol. 163, pp. 485–494, 2019.

[58] I. A. Christensen and S. Schiaffino, "Social influence in group recommender systems," *Online Information Review*, vol. 38, no. 4, pp. 524–542, 2014.

Hindawi
Complexity
Volume 2021, Article ID 8817667, 15 pages
https://doi.org/10.1155/2021/8817667

Research Article

The Risk Priority Number Evaluation of FMEA Analysis Based on Random Uncertainty and Fuzzy Uncertainty

Xiaojun Wu and Jing Wu

School of Economics and Management, Tongji University, Shanghai 200092, China

Correspondence should be addressed to Jing Wu; wujing@tongji.edu.cn

Received 14 August 2020; Revised 13 January 2021; Accepted 6 February 2021; Published 26 February 2021

Academic Editor: Shangce Gao

The risk priority number (RPN) calculation method is one of the critical subjects of failure mode and effects analysis (FMEA) research. Recently, RPN research under a fuzzy uncertainty environment has become a hot topic. Accordingly, increasing studies have ignored the important impact of the random sampling uncertainty in the FMEA assessment. In this study, a fuzzy beta-binomial RPN evaluation method is proposed by integrating fuzzy theory, Bayesian statistical inference, and the beta-binomial distribution. This model can effectively realize real-time, dynamic, and long-term evaluation of RPN under the condition of continuous knowledge accumulation. The major contribution of the proposed model is to use the random uncertainty and fuzzy uncertainty in an integrated model and provide a Markov Chain Monte Carlo (MCMC) method to solve the complex integrated model. The study presented a case study, which presented how to apply this model in practice and indicated the significant influence on the measurement error caused by ignoring the random uncertainty caused by expert evaluation in RPN calculations.

1. Introduction

FMEA was initially developed as a formal design methodology in the 1960s in the aerospace industry [1]. At present, it has been widely adopted to improve the security and reliability of systems [2] and for continuous improvements in product or process design [3] in various fields, for instance, wind power [4], food [5], healthcare [6], fabrics [7], construction [8], healthcare [9], and mining [10].

The traditional FMEA analysis comprises five steps. First, a group of experts are to identify all possible potential failure modes of the product or system. Second, three risk factors are taken into account for each potential failure mode: the occurrence/probability of the failure (O), the severity of the consequences (S), and the chance/probability of the failure going undetected (D) [11]. In the third step, the three factors of severity, occurrence, and detectability are multiplied together to calculate the so-called risk priority number (RPN): $\mathrm{RPN} = S \times O \times D$. Fourth, the critical failure modes are identified based on RPN rankings. Finally, the continuous improvement activities are implemented to reduce the risk of failure modes.

Three risk factors were assessed using a 10-point scale to obtain RPNs for potential failure modes. For more information, please refer to the article [12].

Although the FMEA has been studied for nearly 60 years, the theory and method still have many shortcomings (see Literature Review). Thus, researchers have attempted to improve the traditional FMEA method from various aspects and have adopted different methods to make it more adaptable. The most important extension is the studies that consider the linguistic fuzzy uncertainty of expert evaluation in the RPN calculation. However, few studies have examined the effect of random uncertainty and linguistic fuzzy uncertainty in expert assessment on the results of FMEA evaluation when they act together.

This study presents a novel method that incorporates fuzzy and probabilistic theories to compute RPN in fuzzy and stochastic uncertainty environments. It regards the evaluation process of all experts for a specific factor (such as factor S) as a stochastic process that conforms to the beta-binomial distribution with $n = 10$. Furthermore, an expert's scoring result for this specific factor comes from a fuzzy linguistic evaluation. In this way, the fuzzy uncertainty and

random uncertainty in the expert scoring process can be considered simultaneously to establish an integrated evaluation method.

The major contributions of the study are (1) to enrich the research literature on FMEA by considering both stochastic and fuzzy uncertainties simultaneously; (2) a fuzzy beta-binomial distribution evaluation method is proposed to precisely describe the experts' evaluation; (3) this study presents a method for solving complex RPN models containing random uncertainty and fuzzy uncertainty with Markov Chain Monte Carlo (MCMC) method. The theoretical and practical contributions of the study are discussed in detail in the final section.

The rest of this paper is organized as follows. Section 2 describes the main shortcomings of traditional FMEA and its improvements. In Section 3, we explain the theoretical underpinnings of the study, such as the linguistic fuzzy method, Bayes' theorem, and the beta-binomial distribution. Section 4 introduces the integrated approach, and Section 5 gives a case study. The results of the case study are discussed in detail in Section 6. Finally, we draw conclusions and make recommendations for future research in Section 7.

2. Literature Review

2.1. Traditional FMEA and Its Improvements. At present, FMEA is usually used in some service and manufacturing sectors to eliminate failures and potential problems by evaluating failure modes of new or existing products, processes, or systems [13].

Among all the review articles on FMEA research, one of the most representative articles [12] summarizes the main findings of FMEA during 2009–2012. Many scholars [14–17] have since reviewed the development of FMEA research in recent years. The major shortcomings of the traditional FMEA method [8, 12, 17–19] are shown in Table 1. In response to these shortcomings, researchers have proposed many improved methods, which are also shown in Table 1.

Five major categories were used in the article [12]: multicriteria decision making (MCDM), artificial intelligence (AI), mathematical programming, hybrid approaches, and others. Many later scholars [17] have also used this classification. By contrast, this study starts from the nine shortcomings of the traditional FMEA and gives a brief literature review.

To solve the first and second shortcomings of traditional FMEA research, scholars have improved the traditional FMEA method by adding weights to the three factors. In the article [28], the data envelopment analysis (DEA) was used in the study to determine the relative importance of risk factor weights. The study [20] used ordered weights and proposed a method to reprioritize the failure modes in FMEA by combining fuzzy OWA and DEMATEL methods. Similar studies also include the articles [2, 18, 22]. Fuzzy theory has recently attracted increasing attention in weight determination research. For instance, the intuitionistic fuzzy weighted averaging (IFWA) operator [21], the fuzzy AHP method in [29], the triangular intuitionistic fuzzy entropy method [30], the three-dimensional geometric approach to fuzzy weighted Euclidean (FWE) FMEA, the IVIF MULTIMOORA method [31], and even the integrated method by extended fuzzy AHP and fuzzy MULTIMOORA [32] are all these types of typical studies. Cost [4, 6, 7, 24], customer demand [33, 34], and quality [7] are integrated into FMEA to solve the third shortcoming in many studies. For the fourth shortcoming of traditional FMEA, the Bayesian network (BN) method [16, 35–39] is the most common solution, and other methods include fuzzy cognitive maps [1], FTA analysis [25], and DEMATEL [40–42]. To address the fifth shortcoming, scholars have proposed MCDM to evaluate the potential failure based on three criteria to avoid direct calculation of the RPN. The typical methods include TOPSIS [5, 26, 43], VIKOR [27, 44], MULTIMOORA [31], and DEA [45].

The main extensions of traditional FMEA take uncertainty into consideration to solve the sixth shortcoming. The following section presents a detailed explanation.

2.2. FMEA Research Based on Uncertainty Theory. Almost all systems cannot capture information perfectly, and some of the available information is uncertain due to limited knowledge and cognition [40]. In all kinds of engineering problems, uncertainty is inevitable [13]. Therefore, it is widely believed that risk factors O, S and D are not easy to be used for accurate evaluation [46]. Thus, in recent years, GRA, fuzzy set theory, rough set theory, Dempster-Shafer theory, and probability theory have been used in FMEA research.

2.2.1. Grey Relational Analysis (GRA). Grey relational analysis (GRA) is an important application of grey system theory pioneered by Professor Deng in 1982. GRA is a widely used uncertainty method and has been used to study the FMEA factors. For instance, the integrated GRA and the DEMATEL method to rank the risk of failure were proposed in the study [40]. MCDM in combination with grey theory was studied [18]. In another paper [47], the uncertain information called D numbers and an improved GRA method were proposed for risk evaluation in FMEA.

2.2.2. Fuzzy Set Theory. Because human beings are more accustomed to the direct usage of language variables [48], big efforts have been taken to evaluate risk factors in a linguistic manner [46, 49]. The imprecise, vague, or partially true information was addressed by using the fuzzy set theory [14]. There are so many studies [3, 50, 51] that have combined FMEA with fuzzy sets to handle the weaknesses of the traditional RPN method.

2.2.3. Integrated Fuzzy Methods. Many new methods for expanding FMEA research have emerged from combining fuzzy set theory with other research methods. For instance, the combination of fuzzy set theory and GRA is typical [52–54]. In addition, fuzzy MULTIMOORA [32], fuzzy DEMATEL [55], and an intuitionistic fuzzy approach [30]

Table 1: Deficiencies of traditional FMEA and measures to improve it.

No.	The shortcoming of the traditional FMEA	Improvements and representative articles
1	The relative importance between O, S and D was not considered. It is assumed that these three factors are of equal importance, but this may not be the case when considering the practical application of FMEA.	Weights are assigned to three factors based on various weighting methods, such as OWA [20], IFWA [21], BWM [22], and FWE [23].
2	Different O, S and D rating sets may produce exactly the same RPN values, but their hidden risk implications may be completely different. This issue may result in wasted resources and time, or, in some cases, high-risk failure modes were not widely known.	The introduction of factor weights reduces and avoids the confusion caused by the same RPN results in different failure modes.
3	RPN calculation considers only three risk factors, mainly safety, and ignores other important factors such as quality and cost.	Cost [4], quality [7], and other factors [24] are added to improve the theoretical basis of the RPN evaluation.
4	The RPN approach does not consider the direct/indirect relationship between failure modes and is flawed for systems with many subsystems and components. When one failure causes several other failure modes, that failure should be prioritized for corrective action.	The FTA [25], Bayesian network [16], and other methods are used to present the interactions and relationships of various failures.
5	The three risk factors O, S, and D are evaluated on a discrete ordinal scale. However, the multiplication is not meaningful on the ordinal scale. Thus, the results obtained are not only meaningless, but also in fact misleading.	Few articles discuss the ordinal scale and multiplication issues. Alternatively, MCDM methods, such as TOPSIS [26] and DEMATEL [27], are used to prioritize the failure modes directly.
6	The three risk factors are often difficult to determine accurately. FMEA team members often provide different types of assessment information for the same risk factor, and some of the assessment information may be inaccurate, uncertain, and incomplete due to time constraints, inexperience, and insufficient data.	Introduce uncertainty assessment methods, such as fuzzy theory, rough theory, evidence theory, and probability theory into the FMEA analysis (see Section 2.2).
7	The mathematical form used to calculate RPN is very sensitive to changes in the assessment of risk factors.	Few articles discuss this issue.
8	The rating transitions for the three components of the FM are different. The relationship between the probability table for O and O is nonlinear, whereas the relationship between the probability table for $D(S)$ and $D(S)$ is linear.	Few articles discuss this issue.
9	The results of RPNs are discrete, and many holes are there.	Few articles discuss this issue.

are proposed to evaluate FMEA. In another interesting study [56], S and D are obtained from fuzzy rules, and O is obtained from an artificial neural network.

Although popular, many controversies about the use of the fuzzy set theory method exist. For instance, appropriate member functions for risk factors and priorities are difficult to define [12]. In fuzzy set processing, when defuzzification is done to calculate the final ordering of the failure modes, decisions based on the crisp analogues of a fuzzy set ignore the entropy [50]. Therefore, such rule-based approaches are often too subjective, expensive, and time-consuming and may not be the best method [17].

2.2.4. Rough Set Theory. Rough sets are another important method for studying uncertainty. The studies [57, 58] integrated the rough set theory and the TOPSIS to evaluate the risk of failure modes. Another study [59] integrates the rough set theory and the cloud model theory in FMEA analysis.

2.2.5. Dempster-Shafer Theory (DST). Recently, an increasing number of researchers have applied DST to FMEA. In the study [13], a risk-based fuzzy evidential approach was put forward to use the interval-valued DST and fuzzy axiomatic design to assess the risk of failure modes. Similar studies also include [54, 60].

2.2.6. Probability Theory. The BN is an uncertainty reasoning method that is most effective for failure structure analysis with a discrete probability table. The method does not solve the randomness of finite sampling from a population. Specifically, in FMEA analysis, the evaluation of a factor by experts is a sampling process. Although probabilistic risk analysis is an important topic in quality management and reliability research, random uncertainty is widely ignored by FMEA researchers.

On the other hand, the FMEA results remain static and are not updated with new failure knowledge [61]. Bayesian inference provides a way to dynamically evaluate RPN. In the study [62], based on the FMEA results, the cartridge and mechanical parts components are identified as principal contributors and analyzed meticulously. Data processing techniques within the framework of Bayesian inference are developed to achieve data format consistency in data aggregation and to perform uncertainty analysis using the Monte Carlo approach. Similar studies also include [8].

Furthermore, with the development of the mathematical science, some more instructive methods are put forward to evaluate the risk. In the study [63], a novel uncertain risk

index model is presented, which can be used in RPN calculation.

2.2.7. Z-Number Theory. Since Zadeh [64] proposed to use Z-numbers to express uncertainty in 2011; the Z-numbers theory has gained more and more attention [65, 66]. For example, the linguistic Z-numbers theory [67] combining linguistic term sets and Z-numbers was proposed by Wang et al. and Huang et al. The study [68], combined with Projection Method, proposes a new FMEA model. Similar studies also include [64, 69].

2.3. Summary. The above literature review shows that the shortcomings of the traditional FMEA method are partially solved by the following methods: (1) introduction of factor weights into the RPN calculation; (2) analysis of failure mode relationships and structures; (3) introduction of new factors, such as quality and cost; and (4) fuzzy expression and calculation of RPN factors and weights. However, many remaining problems require additional research. In particular, problems 7, 8, and 9 in Table 1 have received minimal attention. In addition, the linguistic fuzziness of expert scoring has been extensively and intensively studied, but the sampling randomness of expert scoring has not received sufficient attention. Therefore, a fuzzy beta-binomial RPN evaluation method that integrates fuzzy set theory, Bayesian statistical inference, and the beta-binomial distribution is proposed in the following section. This study considers both the randomness and ambiguity in expert evaluation, as well as dynamic changes, to address the shortcomings of traditional FMEA.

3. Preliminaries

Some relevant theories and methods will be introduced briefly in this section in order to derive the subsequent method.

3.1. Linguistic Variables and Defuzzification. In many real situations, data uncertainty comes from two different sources: randomness and vagueness [14]. In FMEA research, the factors O, S, and D are usually evaluated in a linguistic manner [49]. A typical example [19] of using linguistic variables to express O, S, and D is shown in Table 2.

The evaluation results of the three factors O, S, and D expressed by fuzzy numbers should be converted to a defuzzified value. Several methods are used for defuzzification. The centroid method is common, and for a triangular fuzzy number $\widetilde{a} = (a_1, a_2, a_3)$, the defuzzified value $\overline{x}_0(\widetilde{a})$ is expressed by the following equation [19]:

$$\overline{x}_0(\widetilde{a}) = \frac{1}{3}(a_1 + a_2 + a_3). \tag{1}$$

Consider a fuzzy set $\widetilde{A}$. The α-cut set can be denoted by $\widetilde{A}[\alpha]$, where $0 \le \alpha \le 1$. The set $\widetilde{A}[\alpha]$ is a crisp set called the α-cut set of fuzzy set $\widetilde{A}$.

Table 2: Linguistic variables for rating failure modes.

Linguistic variable	Triangular fuzzy number
Very low (VL)	(0, 0, 1)
Low (L)	(0, 1, 3)
Medium low (ML)	(1, 3, 5)
Medium (M)	(3, 5, 7)
Medium high (MH)	(5, 7, 9)
High (H)	(7, 9, 10)
Very high (VH)	(9, 10, 10)

This table is reproduced from [19], [under the Creative Commons Attribution License/public domain].

$$\widetilde{A}[\alpha] = \{x \mid \mu_A(x) \ge \alpha\}, \quad 0 \le \alpha \le 1, \tag{2}$$

$\mu_A(x)$ expresses the membership function of $\widetilde{A}$. For triangular fuzzy numbers, $\widetilde{A}[1] = a_2$ and $\widetilde{A}[0] = [a_1, a_3]$.

3.2. Bayes' Theorem and Posterior Distribution. In FMEA research, the evaluation results of the three factors S, O, and D are usually obtained by expert scoring, and each evaluation result is a discrete value distributed in the interval of 0–10 (for a fuzzy related study, it may be 0–10, as shown in Table 2). This evaluation result is analogous to the beta-binomial distribution with $n = 10$. For example, an evaluation result of 3 is analogous to the success of 3 of 10 Bernoulli experiments. The binomial distribution with parameters n and θ is the discrete probability distribution of the number of successes z in a sequence of n independent Bernoulli experiments, and θ is the probability of success for each trial. The formula of the binomial distribution is

$$P(X = z \mid \theta, n) = \binom{n}{z} \theta^z (1-\theta)^{n-z}. \tag{3}$$

Since each expert is different, the θ values may differ to illustrate experts' uniqueness. If θ is not a fixed value but a random variable that conforms to the beta distribution, then the distribution is a beta-binomial distribution. The letters a and b are parameters of the Beta distribution.

$$p(\theta) = \text{Beta}(a, b) = \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{B(a,b) \propto \theta^{(a-1)}(1-\theta)^{(b-1)}}, \tag{4}$$

where $B(a, B)$ is a simple normalized constant, which can ensure that the area under the beta density integrates to one. In other words, the normalizer for the beta distribution is the beta function.

$$B(a, b) = \int_0^1 \mathrm{d}\theta \theta^{(a-1)}(1-\theta)^{(b-1)}. \tag{5}$$

Therefore, we can use the analogy between the beta-binomial distribution and expert scoring to express the randomness of the expert scoring process.

Since the characteristics of each expert are not known in advance, the a priori information of $p(\theta)$ is lacking. Given a sufficiently large amount of data, two or more Bayesian models with different priors will tend to

converge to the same result. If we have no prior knowledge, we can use flat priors that do not convey much information. Therefore, the prior distribution of such information is Beta $(\alpha = 1, \beta = 1)$. After expert scoring, we obtain new information about each factor, and we can use Bayesian inference to update the knowledge and obtain the posterior probability $p(\theta \mid z, N)$. The relationship between the prior distribution and the posterior distribution can be expressed by Bayes' theorem:

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}. \tag{6}$$

When $p(\theta) = \text{Beta}(a, b)$ and $p(y \mid \theta)$ is a binomial distribution, the posterior distribution calculation process is as follows:

$$\begin{aligned} p(\theta \mid z, n) &= \frac{p(z, n \mid \theta)p(\theta)}{p(z, n)} \\ &= \frac{\binom{n}{z}\theta^z(1-\theta)^{(n-z)}\left(\theta^{(a-1)}(1-\theta)^{(b-1)}/B(a,b)\right)}{p(z,n)} \\ &= \binom{n}{z}\frac{\theta^z(1-\theta)^{(N-z)}\theta^{(a-1)}(1-\theta)^{(b-1)}}{p(z,n)B(a,b)} \\ &= \binom{n}{z}\frac{\theta^{(z+a-1)}(1-\theta)^{((N-z+b)-1)}}{p(z,n)B(a,b)} \\ &= \binom{n}{z}\frac{\theta^{(z+a-1)}(1-\theta)^{((n-z+b)-1)}}{B(z+a, n-z+b)} \\ &\propto \text{Beta}(z+a, n-z+b). \end{aligned} \tag{7}$$

Therefore, the posterior distribution $p(\theta \mid z, n)$ is a beta distribution with parameters $(z + a, n - z + b)$. In this way, the information about θ is updated, and we can further calculate the beta-binomial distribution with the updated θ.

3.3. Beta-Binomial Distribution. In the case of $p(\theta) = \text{Beta}(a, b)$, the beta-binomial distribution is denoted as

$$f(k \mid nt, naq, hb) \sim \text{Beta} \sim \text{binomial}(n, a, b). \tag{8}$$

This formula indicates that the Bernoulli experiment is performed n times, and the probability of k successes is obtained.

$$\begin{aligned} f(k \mid n, a, b) &= \int_0^1 L(\theta \mid k)p(\theta \mid a, b)\mathrm{d}\theta \\ &= \binom{n}{k}\theta^k(1-\theta)^{n-k} \cdot \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{B(a,b)} \\ &= \binom{n}{k}\frac{B(k+a, n-k+b)}{B(a,b)} \end{aligned} \tag{9}$$

where

$$\begin{aligned} p(\theta \mid a, b) &= \text{Beta}(a, b) \\ &= \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{B(a,b)}, \quad \text{for } 0 \le \theta \le 1, \end{aligned} \tag{10}$$

$$P(X = k \mid \theta, n) = L(\theta \mid k) = \binom{n}{k}\theta^k(1-\theta)^{n-k}.$$

A new posterior distribution is plugged into (9).

$$f(k \mid n, a, b) \sim \text{Beta} \sim \text{binomial}(n, k + z + a, N - z - k + b). \tag{11}$$

That is, the probability of the score k of a certain factor is consistent with the above distribution, where $n = 10$, $N =$ expert number $\times 10$, and z is the sum of all experts' evaluation scores for a certain factor.

3.4. The Operation Laws of Fuzzy Power. The operation laws of fuzzy power numbers are shown as follows:

$$\begin{aligned} \theta^{\tilde{z}}\theta^{\tilde{y}} &= \theta^{\tilde{z}+\tilde{y}}, \\ \frac{\theta^{\tilde{z}}}{\theta^{\tilde{y}}} &= \theta^{\tilde{z}-\tilde{y}}, \quad \theta \neq 0, \\ \left(\theta^{\tilde{z}}\right)^{\tilde{y}} &= \theta^{\tilde{z}\cdot\tilde{y}}, \\ (\alpha \cdot \beta)^{\tilde{z}} &= \alpha^{\tilde{z}}\beta^{\tilde{z}}. \end{aligned} \tag{12}$$

The $\tilde{z}$ and $\tilde{y}$ are fuzzy numbers. These operation laws are referred to when the mathematical reasoning is done for the fuzzy beta-binomial distribution in the following section.

4. The Proposed Method

4.1. Fuzzy Beta-Binomial Approach for SOD Evaluation. When the experts score the *S*, *O* and *D* factors separately, the results are usually ambiguous and can be expressed as fuzzy numbers due to the use of linguistic variables. As mentioned earlier, the *S*, *O*, *D* scoring process can be analogized to a binomial experiment with $n = 10$, and the expert scoring result z is a fuzzy number $\widetilde{z}$.

$$\widetilde{P}(X = \widetilde{z} \mid \theta, n) = \binom{n}{z} \theta^{\widetilde{z}} (1-\theta)^{n-\widetilde{z}}, \quad 0 \le \alpha \le 1. \tag{13}$$

Based on the fuzzy algebra, we obtain

$$\widetilde{P}(X = \widetilde{z} \mid \theta, n)[\alpha] = \left\{ \binom{n}{z} \theta^{\widetilde{z}} (1-\theta)^{n-\widetilde{z}} \right\}. \tag{14}$$

If $\widetilde{P}(X = \widetilde{z} \mid \theta, n)[\alpha] = [P_{z1}(\alpha), P_{z2}(\alpha)]$ then

$$P_{z1}(\alpha) = \min \left\{ \binom{n}{z} \theta^{\widetilde{z}} (1-\theta)^{n-\widetilde{z}} \right\}, \tag{15}$$

and

$$P_{z2}(\alpha) = \max \left\{ \binom{n}{z} \theta^{\widetilde{z}} (1-\theta)^{n-\widetilde{z}} \right\}. \tag{16}$$

In this case, $\widetilde{z}$ is expressed as a fuzzy linguistic data shown as in Tables 3 and 4. For any specific expert, the value of $\widetilde{z}$ can be substituted into the formula of $P_{z1}(\alpha)$ and $P_{z2}(\alpha)$ to calculate $\widetilde{P}(X = \widetilde{z} \mid \theta, n)$.

The binomial distribution is a likelihood function. Based on the prior distribution $\text{Beta}(a, b)$ and binomial likelihood functions, the posterior distribution of the expert evaluation can be written as

$$\begin{aligned}
\widetilde{p}(\theta \mid z, N) &= \frac{\widetilde{p}(\widetilde{z}, tN \mid n\theta) p(\theta)}{\widetilde{p}(\widetilde{z}, tN)} \\
&= \frac{\binom{n}{\widetilde{z}} \theta^{\widetilde{z}} (1-\theta)^{n-\widetilde{z}} \left(\theta^{(a-1)} (1-\theta)^{(b-1)} / B(a,b) \right)}{\widetilde{p}(\widetilde{z}, N)} \\
&= \binom{n}{\widetilde{z}} \frac{\theta^{\widetilde{z}} (1-\theta)^{N-\widetilde{z}} \theta^{(a-1)} (1-\theta)^{(b-1)}}{\widetilde{p}(\widetilde{z}, N) B(a,b)} \\
&= \binom{n}{\widetilde{z}} \frac{\theta^{(\widetilde{z}+a-1)} (1-\theta)^{((N-\widetilde{z}+b)-1)}}{\widetilde{p}(\widetilde{z}, N) B(a,b)}.
\end{aligned} \tag{17}$$

Since the conjugate distribution of the beta distribution is a binomial distribution, its posterior distribution is also a beta distribution. From (17), after discarding the irrelevant terms of θ, we obtain

$$\begin{aligned}
&\widetilde{p}(\theta \mid z, N) \\
&\propto \theta^{(\widetilde{z}+tan-q1)} (1-\theta)^{((N-\widetilde{z}+b)-1)} \\
&\propto \text{Beta}(\widetilde{z} + a, N - \widetilde{z} + b).
\end{aligned} \tag{18}$$

Under the posterior distribution θ, the final evaluation results for SOD are

$$f(k \mid n, a', b') = \binom{n}{k} \frac{B\left(k + \widetilde{a}', n - k + \widetilde{b}'\right)}{B(a', b')}, \tag{19}$$

where $n = 10$, $\widetilde{a}' = \widetilde{z} + a$, and $\widetilde{b}' = \widetilde{z} + b$. This result can be rewritten as

$$f(\widetilde{k} \mid n, a, b) \sim \text{Beta} - \text{binomial}(n, \widetilde{z} + k + a, N - \widetilde{z} - k + b). \tag{20}$$

4.2. The Proposed Evaluation Process. The integrated proposed approach based on the fuzzy beta-binomial distribution and Bayesian inference is shown in Figure 1 and described below.

Step 1. An expert group that consists of n people is established to evaluate the SOD factors and obtain the evaluation value of the jth factor by the ith expert. The prior probability distribution is Beta (1, 1).

Step 2. According to Bayes' theorem, combined with the known prior distribution Beta (1, 1) and binomial distribution likelihood function, the posterior distribution obtained is proportional to $\text{Beta}(\widetilde{z} + 1, N - \widetilde{z} + b)$, where $N = 10n$, $\widetilde{z} = \sum_{i=1}^{n} \widetilde{x}_{ij}$.

Step 3. Calculate the theoretical value of the expert evaluation based on the beta-binomial distribution with θ posterior distribution $\text{Beta}(\widetilde{z} + 1, N - \widetilde{z} + 1)$:

$$f(\widetilde{k} \mid n, a, b) \sim \text{Beta} - \text{binomial}(n, \widetilde{z} + k + 1, N - \widetilde{z} - k + 1). \tag{21}$$

Thus, we can obtain SOD factors' theoretical values.

Step 4. By using MCMC, we can take samples from each of the SOD theoretical distribution and calculating the $\text{RPN} = S \times O \times D$; furthermore, the combined distribution of RPN can be gotten. We can prioritize the failure modes based on the combined distribution.

Overall, the first step is to collect expert evaluations of the severity (*S*), occurrence (*O*), and probability of the failure going undetected (*D*) in the form of fuzzy linguistic expressions (see the case study, Tables 3 and 5). In the second step, the expert evaluation is regarded as a random process, and the randomness is presented by using beta-binomial distribution for each of the SOD factors. In the third step, MCMC method is used to get the samples from each of the distributions of *S*, *O* and *D* factors. Furthermore, the RPN can be calculated based on the samples from three beta-binomial distributions. The final priority of each failure is

Table 3: The original data set.

	O					S					D				
—	DM1	DM2	DM3	DM4	DM5	DM1	DM2	DM3	DM4	DM5	DM1	DM2	DM3	DM4	DM5
FM1	M	M	M	MH	M	ML	ML	ML	M	M	M	ML	ML	ML	ML
FM2	H	MH	H	MH	MH	H	MH	H	H	H	M	M	ML	M	M
FM3	VH	MH	VH	VH	VH	MH	MH	MH	MH	MH	MH	M	MH	MH	M
FM4	M	M	L	M	M	M	M	ML	M	M	VL	ML	VL	ML	VL
FM5	M	ML	M	M	M	M	MH	MH	M	M	L	ML	L	L	L
FM6	MH	H	M	MH	M	H	H	H	H	H	L	M	L	L	VL
FM7	ML	L	ML	ML	ML	VH	H	H	VH	H	VL	VL	VL	L	VL

Note: DM1 means the first expert's measurement. DM2, DM3, DM4, DM5 are in the same fashion. This table is reproduced from [19], [under the Creative Commons Attribution License/public domain].

determined by comparing the parameters of the RPNs of each of the failures.

5. Case Study

The following section is a case study using four different approaches shown in Table 5 and the data set used in the article [19]. The original evaluation data set is shown in Table 3, in which there are seven failure modes and five evaluation experts.

According to the linguistic variables in Table 2, the original data set can be further expressed in fuzzy numbers, as shown in Table 4.

After calculating the θ posterior distribution, the SOD factors' fuzzy numbers of each failure mode under α-cut $\alpha = 1$ are obtained, and the corresponding beta-binomial distribution is shown in Table 6.

After sampling (number of samples = 20000), the RPN of each of the failure modes is calculated under $\alpha = 1$ and shown in Table 7.

After sampling (number of samples = 50000), the RPN result of each failure mode under $\alpha = 0$ is calculated and shown in Table 8.

A comparison with the evaluation results of the literature [19] is shown in Table 9.

The detailed comparisons of four approaches are in the following section.

6. Discussion

It can be seen from Table 8 that the ranking of the proposed method differs from that of traditional FMEA and fuzzy TOPSIS for only two failure modes, but the ranking of Liu's model approach differs considerably from the results of the proposed method. This difference is due to two main reasons. First, the proposed method and the traditional FMEA method are based on the result of RPN = $S \times O \times D$, whereas Liu's model and fuzzy TOPSIS are based on an MCDM method. The second reason is the effect of weights on the results in the calculation process.

The only difference in the results of the proposed method and the traditional FMEA is the ranking of FM4 and FM7 because the traditional FMEA method's SOD factors have a scoring range of 1–10, whereas the fuzzy set data may have a value of 0. In addition, the traditional FMEA method calculates the SOD factor scores (generally taking a positive integer). In the calculation process, the data are not forced to take an integer value (for example, a fractional part may exist after the mean is calculated). The above two reasons make the calculation results of FM4 and FM7 slightly different after considering randomness and ambiguity.

The proposed approach is based on RPN = $S \times O \times D$, which conforms to the traditional FMEA method, and the understanding is simple and easy. Furthermore, because O is the probability of the failure, S is the severity of the failure, and D is the probability of not detecting the failure, although SOD is not a true probability value, the values can be converted by means of a specific method from the original data. A one-to-one correspondence exists with the actual probability values, so there is a function mapping relationship between the evaluation value of the SOD and the real value. The SOD data of different dimensions and measurement scales are uniformly converted into discrete values in 1–10, just as the data of different dimensions and measurement scales are standardized and converted into data in the interval 0–1. This is simply a process of data standardization; therefore, the direct multiplication of O and D after standardization has practical meaning. If the S factor is regarded as a weighting factor of failure severity, it can be multiplied by O and D to express the severity of a failure mode. The result of the multiplication of S and $O \times D$ expresses the urgency of improving the failure based on the failure probability and detection difficulty of a certain failure mode. Thus, the calculation of $S \times O \times D$ also has practical meaning. Although there are many criticisms of the method of sorting failure mode priorities by calculating RPN = $S \times O \times D$ for discrete ordinal scales of measure, the basic idea of resorting RPN by multiplying the three factors has a strong theoretical basis. In terms of the practical meaning of RPN = $S \times O \times D$, the additional physical weight of each factor is not reasonable in some sense.

Fuzzy uncertainty and random uncertainty are considered in the evaluation of the SOD factors of each failure mode. From the calculation results, the fuzzy numbers of each factor are consistent with the ranking results of $\alpha = 1$ and $\alpha = 0$. From the perspective of a simplified calculation in the future, it is sufficient to do the calculation considering only $\alpha = 1$. Moreover, the graph shows that the seven failure modes can be divided into three groups. The distribution patterns of FM2 and FM3 are significantly different from

Table 4: Fuzzy expression of the original data set.

	O					S					D				
—	DM1	DM2	DM3	DM4	DM5	DM1	DM2	DM3	DM4	DM5	DM1	DM2	DM3	DM4	DM5
FM1	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(5, 7, 9)	(3, 5, 7)	(1, 3, 5)	(1, 3, 5)	(1, 3, 5)	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(1, 3, 5)	(1, 3, 5)	(1, 3, 5)	(1, 3, 5)
FM2	(7, 9, 10)	(5, 7, 9)	(7, 9, 10)	(5, 7, 9)	(5, 7, 9)	(7, 9, 10)	(5, 7, 9)	(7, 9, 10)	(7, 9, 10)	(7, 9, 10)	(3, 5, 7)	(3, 5, 7)	(1, 3, 5)	(3, 5, 7)	(3, 5, 7)
FM3	(9, 10, 10)	(5, 7, 9)	(9, 10, 10)	(9, 10, 10)	(9, 10, 10)	(5, 7, 9)	(5, 7, 9)	(5, 7, 9)	(5, 7, 9)	(5, 7, 9)	(5, 7, 9)	(3, 5, 7)	(5, 7, 9)	(5, 7, 9)	(3, 5, 7)
FM4	(3, 5, 7)	(3, 5, 7)	(0, 1, 3)	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(1, 3, 5)	(3, 5, 7)	(3, 5, 7)	(0, 0, 1)	(1, 3, 5)	(0, 0, 1)	(1, 3, 5)	(0, 0, 1)
FM5	(3, 5, 7)	(1, 3, 5)	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(3, 5, 7)	(5, 7, 9)	(5, 7, 9)	(3, 5, 7)	(3, 5, 7)	(0, 1, 3)	(1, 3, 5)	(0, 1, 3)	(0, 1, 3)	(0, 1, 3)
FM6	(5, 7, 9)	(7, 9, 10)	(3, 5, 7)	(5, 7, 9)	(3, 5, 7)	(7, 9, 10)	(7, 9, 10)	(7, 9, 10)	(7, 9, 10)	(7, 9, 10)	(0, 1, 3)	(3, 5, 7)	(0, 1, 3)	(0, 1, 3)	(0, 0, 1)
FM7	(1, 3, 5)	(0, 1, 3)	(1, 3, 5)	(1, 3, 5)	(1, 3, 5)	(9, 10, 10)	(7, 9, 10)	(7, 9, 10)	(9, 10, 10)	(7, 9, 10)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	(0, 1, 3)	(0, 0, 1)

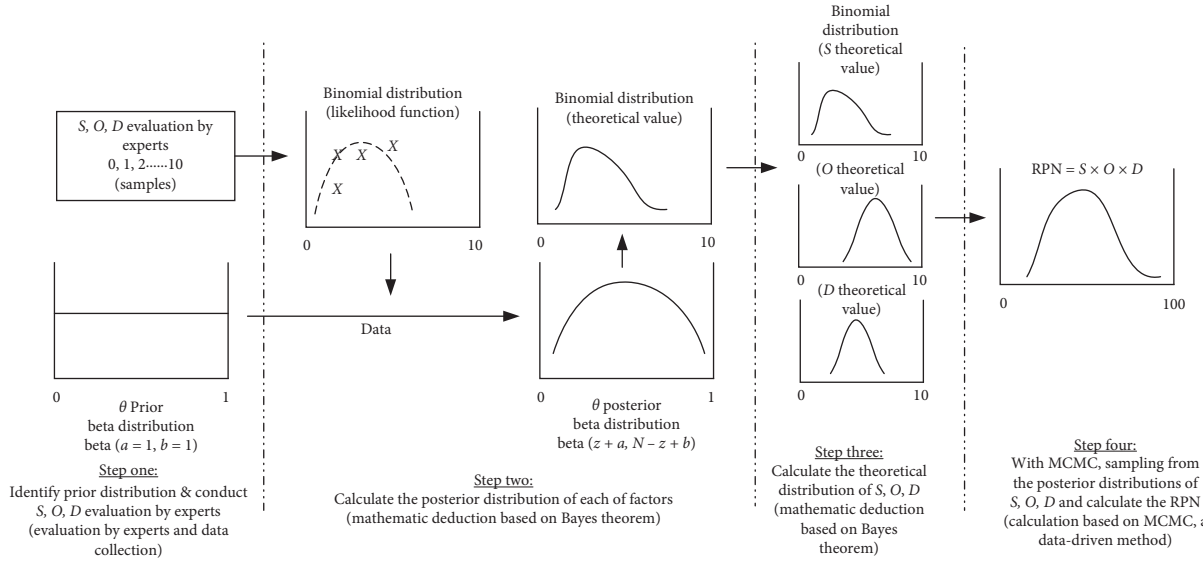


Figure 1: The evaluation process.

Table 5: The comparison between different methods.

Method or model	Expert evaluation	RPN calculation
The proposed method	Fuzzy linguistic expression and regard it as a random process	To express S, O, D with three different beta-binomial distributions; MCMC is used to get samples from three distribution and calculate the corresponding RPN
Liu's model	Fuzzy linguistic expression	Defuzzification, convert the fuzzy expressions to crisp ones and calculate the RPN
Traditional FMEA	Crisp expression	To calculate the RPN directly
Fuzzy TOPSIS	Fuzzy linguistic expression	Defuzzification, convert the fuzzy expressions to crisp ones and use the TOPSIS to evaluate the priorities of various failures

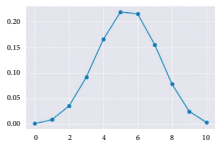
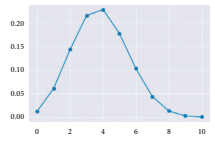
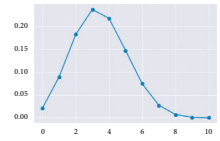
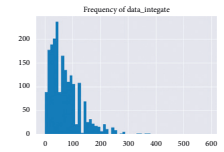
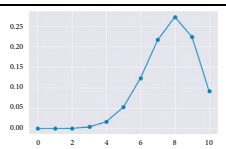
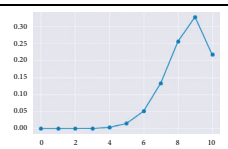
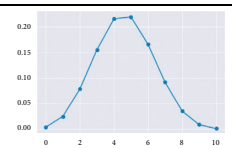
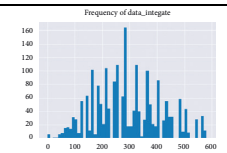
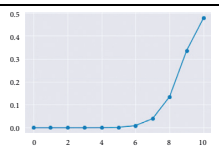
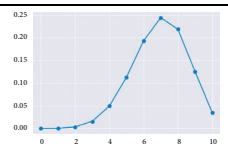
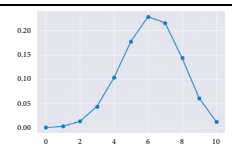
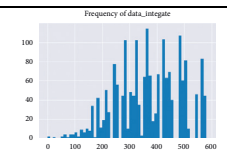
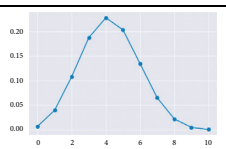
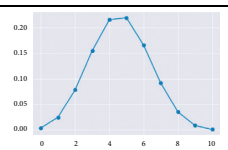
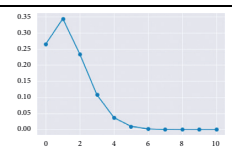
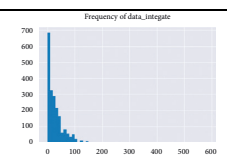
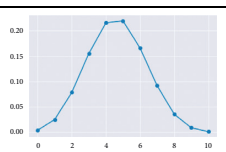
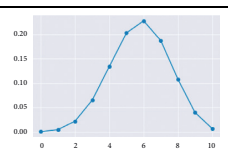
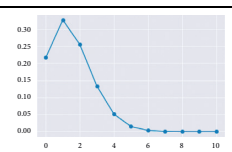
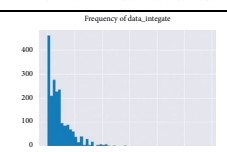
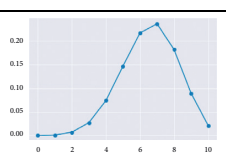
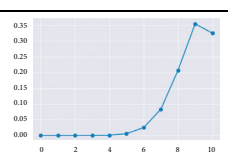
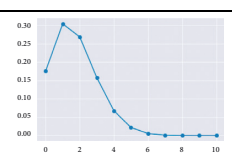
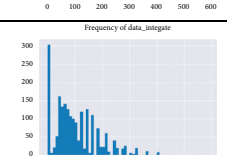
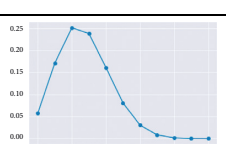
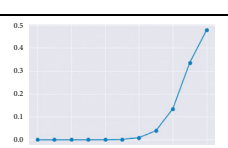
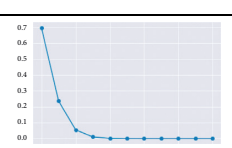
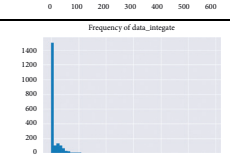
Table 6: The fuzzy beta-binomial distribution with $\alpha = 1$.

	O	$\widetilde{O}[1]$	$\widetilde{p}(\theta\|z,N)[1]$	S	$\widetilde{S}[1]$	$\widetilde{p}(\theta\|z,N)[1]$	D	$\widetilde{D}[1]$	$\widetilde{p}(\theta\|z,N)[1]$
FM1	(17, 27, 37)	$\widetilde{O}[1] = [27]$	Beta-binomial (28, 24)	(9, 19, 29)	$\widetilde{S}[1] = [19]$	Beta-binomial (20, 32)	(7, 17, 27)	$\widetilde{D}[1] = [17]$	Beta-binomial (18, 34)
FM2	(29, 39, 47)	$\widetilde{O}[1] = [39]$	Beta-binomial (40, 12)	(33, 43, 49)	$\widetilde{S}[1] = [43]$	Beta-binomial (44, 8)	(13, 23, 33)	$\widetilde{D}[1] = [23]$	Beta-binomial (24, 28)
FM3	(41, 47, 49)	$\widetilde{O}[1] = [47]$	Beta-binomial (48, 4)	(25, 35, 45)	$\widetilde{S}[1] = [35]$	Beta-binomial (36, 16)	(21, 31, 41)	$\widetilde{D}[1] = [31]$	Beta-binomial (32, 20)
FM4	(12, 21, 31)	$\widetilde{O}[1] = [21]$	Beta-binomial (22, 30)	(13, 23, 33)	$\widetilde{S}[1] = [23]$	Beta-binomial (24, 28)	(2, 6, 13)	$\widetilde{D}[1] = [6]$	Beta-binomial (7, 45)
FM5	(13, 23, 33)	$\widetilde{O}[1] = [23]$	Beta-binomial (24, 28)	(19, 29, 39)	$\widetilde{S}[1] = [29]$	Beta-binomial (30, 22)	(1, 7, 17)	$\widetilde{D}[1] = [7]$	Beta-binomial (8, 44)
FM6	(23, 33, 42)	$\widetilde{O}[1] = [33]$	Beta-binomial (34, 18)	(35, 45, 50)	$\widetilde{S}[1] = [45]$	Beta-binomial (46, 6)	(3, 8, 17)	$\widetilde{D}[1] = [8]$	Beta-binomial (9, 43)
FM7	(4, 13, 23)	$\widetilde{O}[1] = [13]$	Beta-binomial (14, 38)	(39, 47, 50)	$\widetilde{S}[1] = [47]$	Beta-binomial (48, 4)	(0, 1, 7)	$\widetilde{D}[1] = [1]$	Beta-binomial (2, 50)

those of the other failure modes. The patterns of FM1 and FM6 are extremely similar, and the average and discrete degree of FM4, FM5, and FM7 are similar.

Overall, the proposed model in the study has two important advantages when compared with Liu's model and the fuzzy TOPSIS approach: (1) defuzzification is no longer used for the processing of fuzzy data, so more information about the original data can be preserved; and (2) even more complex models including more factors and weights can be processed to calculate the RPN. Thus, the proposed model

Table 7: The RPN under $\alpha = 1$.

	Factor O	Factor S	Factor D	RPN	Results	Ranking
FM1					Mean = 72.3 SD = 56.2 Median = 60.0	4
FM2					Mean = 302.1 SD = 135.8 Median = 288	2
FM3					Mean = 397.3 SD = 147.3 Median = 392	1
FM4					Mean = 25.9 SD = 29.8 Median = 18	6
FM5					Mean = 40.2 SD = 41.6 Median = 30	5
FM6					Mean = 102.3 SD = 84.9 Median = 81	3
FM7					Mean = 9.1 SD = 9.1 Median = 0	7

Note: SD means the standard deviation.

provides a more accurate assessment of the four different approaches in the study.

When considering the expert evaluation process as a random sampling process, the basic theory of applied statistics provides a means to evaluate the confidence of expert evaluation results. Subtle changes in the SOD evaluation values of the five experts have a considerable impact on the final RPN calculation results and the final prioritization. One of the most important reasons is that the number of experts in the case study is small. One way to solve this problem is to increase the number of experts to improve the robustness of the final evaluation results. When considering random uncertainties, the instability of the result is more easily explained by the deviation of the population mean and the sample mean. The estimation of the population mean is obtained by calculating the sample mean. When the population standard deviation is unknown, we have

$$\mu = \overline{x} \pm t\frac{s}{\sqrt{n}}, \tag{22}$$

where μ is the population mean, $\overline{x}$ is the sample mean, s is the population standard deviation $s = \sqrt{(\sum (x_i - \overline{x})^2/(n-1))}$, n is the sample size, and t is the standard normal value corresponding to the desired level of confidence. E is the maximum allowable error. The deviation E between the population mean and the sample mean is

$$E = t\frac{s}{\sqrt{n}}. \tag{23}$$

Thus, the number of samples n can be obtained as

$$n = \left(\frac{ts}{E}\right)^2. \tag{24}$$

Table 8: The RPN under $\alpha = 0$.

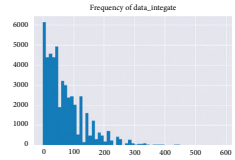
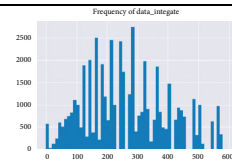
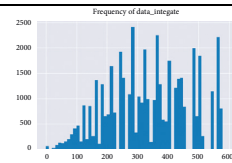
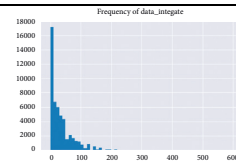
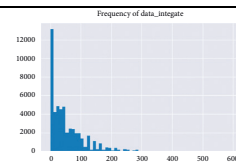
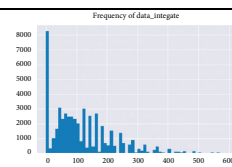
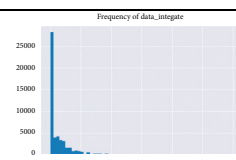
	$\alpha = 0$	RPN	Results	Ranking
FM1	$\tilde{O}[0] = [17, 37]$ $\tilde{S}[0] = [9, 29]$ $\tilde{D}[0] = [7, 27]$		Mean = 72.0 SD = 68.6 Median = 54.0	4
FM2	$\tilde{O}[0] = [29, 47]$ $\tilde{S}[0] = [33, 49]$ $\tilde{D}[0] = [13, 33]$		Mean = 280.4 SD = 156.5 Median = 252.0	2
FM3	$\tilde{O}[0] = [41, 49]$ $\tilde{S}[0] = [25, 45]$ $\tilde{D}[0] = [21, 41]$		Mean = 377.4 SD = 173.7 Median = 360.0	1
FM4	$\tilde{O}[0] = [12, 31]$ $\tilde{S}[0] = [13, 33]$ $\tilde{D}[0] = [2, 13]$		Mean = 32.7 SD = 40.1 Median = 20.0	6
FM5	$\tilde{O}[0] = [12, 33]$ $\tilde{S}[0] = [19, 39]$ $\tilde{D}[0] = [1, 17]$		Mean = 51.4 SD = 57.7 Median = 35.0	5
FM6	$\tilde{O}[0] = [23, 42]$ $\tilde{S}[0] = [35, 50]$ $\tilde{D}[0] = [3, 17]$		Mean = 114.5 SD = 100.7 Median = 90.0	3
FM7	$\tilde{O}[0] = [4, 23]$ $\tilde{S}[0] = [39, 50]$ $\tilde{D}[0] = [0, 7]$		Mean = 21.0 SD = 34.2 Median = 0.0	7

Table 9: Ranking comparison.

	Proposed approach ranking ($\alpha = 1$)	Proposed approach ranking ($\alpha = 0$)	Liu's model ($\phi = 1$)	Liu's model ($\phi = 0$)	Traditional FMEA	Fuzzy TOPSIS
FM1	4	4	6	3	4	3
FM2	2	2	1	2	2	2
FM3	1	1	3	1	1	1
FM4	6	6	7	7	7	6
FM5	5	5	5	5	5	5
FM6	3	3	2	4	3	4
FM7	7	7	4	6	5	7

For example, for the O factor of FM6 in this example (sample $s = 1.673$), within the range of evaluation values 1–10, if $E = 1$, 95% confidence level ($t = 2.776$),

$$n = \left(\frac{ts}{E}\right)^2 = \left(\frac{2.776 \times 1.673}{1}\right)^2 \approx 21.57 \approx 22. \tag{25}$$

Another example is the S factor of FM1 (sample $s = 1.095$) for $E = 1$ and a 95% confidence level ($t = 2.776$):

$$n = \left(\frac{ts}{E}\right)^2 = \left(\frac{2.776 \times 1.095}{1}\right)^2 \approx 9.23 \approx 10. \quad (26)$$

For a 95% confidence level and an E of 1, the number of samples for each failure mode and its evaluation factors are quite different to achieve the required accuracy. In this case, the largest number of samples should be used. The final sample number can guarantee accuracy and allowable error. Taking the O factor of FM6 (sample $s = 1.673$) as an example, the deviation from the theoretical population mean is calculated as follows:

$$E = t\frac{s}{\sqrt{n}} = 2.776 \times \frac{1.673}{\sqrt{5}} \approx 2.07. \quad (27)$$

This result shows that the deviation is too large for the evaluation using a 1–10 scale. If each indicator has such a large deviation, the deviation of the overall RPN $= S \times O \times D$ will be excessive. Even if TOPSIS or VIKOR is used instead of RPN $= S \times O \times D$, the deviation of the results is still large.

The FMEA method considering random uncertainty makes it easier to understand the advantages and disadvantages of FMEA results based on a limited number of expert evaluations. (1) The greater the number of experts that participate in the evaluation, the higher the confidence of the results. (2) The result is the sample mean of a random sampling distribution. The actual population mean and this mean point may not completely coincide. The theoretical SOD factor's true value and the measured SOD factor value must be different. Regardless of whether the traditional FMEA method, fuzzy TOPSIS, or Liu's model is used, if the underlying SOD data have a large deviation from the true value, the accuracy of the results will be questionable. To ensure that the error of the evaluation results is within an acceptable range, the number of experts should be estimated according to formula (24).

One major shortcoming of traditional FMEA research is that the weight of each SOD factor is not considered. As mentioned above, the calculation of the classic RPN $= S \times O \times D$ itself implies a weighting process based on the severity of the failure. Whether including additional weights is appropriate requires further discussion. However, the important point to note is that weights are usually based on expert evaluation. As mentioned above, shortage of experts will substantially weaken the credibility of the results. Improper weighting will increase the risk of incorrect evaluation results. The ranking results of the proposed research method and fuzzy TOPSIS are extremely similar: the only difference is the ranking of FM1 and FM6. However, the difference between the proposed approach and Liu's model is considerable. One of the most important reasons is that the new weight is introduced in the calculation process of the VIKOR method in Liu's model, which dramatically changes the importance of the three SOD factors in the evaluation. The introduction of this new artificial weight vector further increases the deviation between the evaluation result and the theoretical true value of the population.

7. Conclusion

7.1. Summary. Although the traditional FMEA method has many shortcomings, prioritization of potential failure risks by calculating RPN is common in industry because the RPN calculation is simpler and easier to understand than other methods.

As for the actual evaluation process in the real world, there are at least two types of uncertainties, fuzzy uncertainty and random uncertainty. It is more beneficial to build a decision model by considering both uncertainties simultaneously for the improvement the validity of the evaluation results. After deeply understanding the impact of the two uncertainties on the evaluation results, it is possible to more clearly and accurately grasp that uncertainty is the more critical one during the evaluation process, and the result of simplified calculation only focusing on the more critical uncertainty can be closer to the true value we expect to have.

This study proposes a fuzzy beta-binomial distribution evaluation method that integrates triangular fuzzy numbers, Bayesian statistical inference, and the beta-binomial distribution. The fuzzy uncertainty of the evaluation is measured by introducing linguistic variables to make the evaluation process more humanized and more similar to natural language processing. The introduction of the measure of random uncertainty makes the expert scoring process a finite sample sampling process. Because the full sample (or large sample) sampling evaluation cannot be realized, the evaluation of limited experts is inevitably limited by the shortcomings of small sample sizes. The difference between the mean of the samples and the theoretical population mean is often outside the acceptable range, resulting in invalid conclusions, regardless of how sophisticated the data processing is or how complicated the fuzzy expression is. This aspect has been widely ignored by many studies.

In the case study, a total of five experts scored seven failure modes. Formula (25) shows that, under the conditions that error E is 1 and confidence level is 95%, at least 22 experts are required to participate in the evaluation to achieve the required evaluation requirements. However, in the case, only five experts join in the evaluation, and under a 95% confidence level, the error E is 2.07 (as shown in equation (27)), which is far from the requirement. It is found that when the sample size of experts is small, and the evaluation difference of each expert is large, random uncertainty has a greater effect on the evaluation results than fuzzy uncertainty no matter how elaborate the membership function is used in the model.

7.2. The Theoretical Contributions. The main theoretical contributions of this paper are shown as follows.

(1) This study firstly proposes a method to introduce both random and fuzzy uncertainty into FMEA to calculate RPN. It can be seen from the literature review that scholars have extended FMEA studies by introducing fuzzy uncertainty or random uncertainty into traditional FMEA analysis to compensate for the shortcomings of the traditional approach, but

these studies consider either fuzzy or random uncertainty separately and rarely consider both uncertainties simultaneously in a single model.

(2) In order to calculate the RPN with both uncertainties, this study innovatively proposes a method for describing RPN calculations under random and fuzzy uncertainty by using the beta-binomial distributions.

(3) This study pioneers a method for solving complex RPN models containing random uncertainty and fuzzy uncertainty with MCMC method. For the calculation of fuzzy numbers, this is generally achieved by defuzzification, a process that actually loses some of the information of the original equation, whereas a more efficient way to preserve the information of the original equation is through the data-driven method, such as MCMC.

7.3. The Implication to Practice. The main practical contributions of this paper are shown as follows.

(1) In traditional studies, only a single uncertainty is considered, which is not consistent with the actual practical situation. As discussed earlier, when quantifying the three factors of SOD through expert evaluation methods, both random and fuzzy uncertainties do exist, so the RPN calculation method proposed in this study with both random and fuzzy uncertainty methods is in line with the actual situation and has more practical value.

(2) The use of the MCMC method to solve the three complex beta-binomial distributed multiplication problems can effectively reduce the difficulty of solving the complex model, without losing most of the information, and the obtained risk priority evaluation results are more reliable; this method also facilitates the development of automatic solution systems through automotive computer program, thus allowing ordinary quality managers to perform more accurate RPN assessment.

(3) The numerical solution method proposed in this study provides a feasible idea for the further extended solution of the RPN calculation model, such as the calculation after further considering the weighting of the three factors of SOD and the calculation after adding new factors, which are all feasible in this method.

7.4. Limitation and Future Research. Certainly, there are still some problems that need further study: (1) this study does not consider another important dimension of uncertainty, namely, rough uncertainty in the RPN evaluation. Although studies have considered both ambiguity and roughness in the calculation of RPN, the three uncertainties have not been considered simultaneously; (2) the proposed approach uses a beta-binomial distribution to fit the expert evaluation results. In future research, the possibility of using other distributions can be further explored.

Data Availability

No other additional data were used in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. E. Pelaez and J. B. Bowles, "Using fuzzy cognitive maps as a system model for failure modes and effects analysis," *Information Sciences*, vol. 88, no. 1–4, pp. 177–199, 1996.

[2] W. Wang, X. Liu, Y. Qin, and Y. Fu, "A risk evaluation and prioritization method for FMEA with prospect theory and Choquet integral," *Safety Science*, vol. 110, pp. 152–163, 2018.

[3] M. Kumru and P. Y. Kumru, "Fuzzy FMEA application to improve purchasing process in a public hospital," *Applied Soft Computing*, vol. 13, no. 1, pp. 721–733, 2013, in English.

[4] N. Tazi, E. Chatelet, and Y. Bouzidi, "Using a hybrid cost-FMEA analysis for wind turbine reliability analysis," *Energies*, vol. 10, no. 3, p. 276, 2017.

[5] H. Selim, M. G. Yunusoglu, and Ş. Yılmaz Balaman, "A dynamic maintenance planning framework based on fuzzy TOPSIS and FMEA: application in an international food company," *Quality and Reliability Engineering International*, vol. 32, no. 3, pp. 795–804, 2016.

[6] S. A. Rahimi, A. Jamshidi, D. Ait-Kadi, and A. Ruiz, "Using fuzzy cost-based FMEA, GRA and profitability theory for minimizing failures at a healthcare diagnosis service," *Quality and Reliability Engineering International*, vol. 31, no. 4, pp. 601–615, 2015.

[7] T. L. Nguyen, M. H. Shu, and B. M. Hsu, "Extended FMEA for sustainable manufacturing: an empirical study in the non-woven fabrics industry," *Sustainability*, vol. 8, no. 9, p. 939, 2016.

[8] A. Brun and M. M. Savino, "Assessing risk through composite FMEA with pairwise matrix and Markov chains," *International Journal of Quality & Reliability Management*, vol. 35, no. 9, pp. 1709–1733, 2018.

[9] H. C. Liu, *Improved FMEA Methods for Proactive Healthcare Risk Analysis*, pp. 15–45, Springer, Singapore, 2019.

[10] E. Bakhtavar and S. Yousefi, "Assessment of workplace accident risks in underground collieries by integrating a multi-goal cause-and-effect analysis method with MCDM sensitivity analysis," *Stochastic Environmental Research and Risk Assessment*, vol. 32, no. 12, pp. 3317–3332, 2018.

[11] Z. L. Yang, S. Bonsall, and J. Wang, "Fuzzy rule-based Bayesian reasoning approach for prioritization of failures in FMEA," *IEEE Transactions on Reliability*, vol. 57, no. 3, pp. 517–528, 2008.

[12] H.-C. Liu, L. Liu, and N. Liu, "Risk evaluation approaches in failure mode and effects analysis: a literature review," *Expert Systems with Applications*, vol. 40, no. 2, pp. 828–838, 2013, in English.

[13] H. Seiti, A. Hafezalkotob, S. E. Najafi, and M. Khalaj, "A risk-based fuzzy evidential framework for FMEA analysis under uncertainty: an interval-valued DS approach," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1419–1430, 2018.

[14] S. Kabir and Y. Papadopoulos, "A review of applications of fuzzy sets to safety and reliability engineering," *International Journal of Approximate Reasoning*, vol. 100, pp. 29–55, 2018.

[15] C. Spreafico, D. Russo, and C. Rizzi, "A state-of-the-art review of FMEA/FMECA including patents," *Computer Science Review*, vol. 25, pp. 19–28, 2017.

[16] S. Kabir and Y. Papadopoulos, "Applications of Bayesian networks and petri nets in safety, reliability, and risk assessments: a review," *Safety Science*, vol. 115, pp. 154–175, 2019.

[17] U. Asan and A. Soyer, "Failure mode and effects analysis under uncertainty: a literature review and tutorial," in *Intelligent Decision Making in Quality Management: Theory and Applications, Volume 97*, C. Kahraman and S. Yanik, Eds., Springer, Cham, Switzerland, pp. 265–325, 2016.

[18] H.-W. Lo and J. J. H. Liou, "A novel multiple-criteria decision-making-based FMEA model for risk assessment," *Applied Soft Computing*, vol. 73, pp. 684–696, 2018.

[19] H.-C. Liu, J.-X. You, X.-Y. You, and M.-M. Shan, "A novel approach for failure mode and effects analysis using combination weighting and fuzzy VIKOR method," *Applied Soft Computing*, vol. 28, pp. 579–588, 2015, in English.

[20] K.-H. Chang and C.-H. Cheng, "Evaluating the risk of failure using the fuzzy OWA and DEMATEL method," *Journal of Intelligent Manufacturing*, vol. 22, no. 2, pp. 113–129, 2011, in English.

[21] H.-C. Liu, L. Liu, and P. Li, "Failure mode and effects analysis using intuitionistic fuzzy hybrid weighted Euclidean distance operator," *International Journal of Systems Science*, vol. 45, no. 10, pp. 2012–2030, 2014.

[22] J. Rezaei, "Best-worst multi-criteria decision-making method," *Omega*, vol. 53, pp. 49–57, 2015.

[23] J. Park, C. Park, and S. Ahn, "Assessment of structural risks using the fuzzy weighted Euclidean FMEA and block diagram analysis," *International Journal of Advanced Manufacturing Technology*, vol. 99, no. 9–12, pp. 2071–2080, 2018.

[24] M. J. Rezaee, A. Salimi, and S. Yousefi, "Identifying and managing failures in stone processing industry using cost-based FMEA," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 9–12, pp. 3329–3342, 2017.

[25] J. F. W. Peeters, R. J. I. Basten, and T. Tinga, "Improving failure analysis efficiency by combining FTA and FMEA in a recursive manner," *Reliability Engineering & System Safety*, vol. 172, pp. 36–44, 2018.

[26] B. Vahdani, M. Salimi, and M. Charkhchian, "A new FMEA method by integrating fuzzy belief structure and TOPSIS to improve risk evaluation process," *International Journal of Advanced Manufacturing Technology*, vol. 77, no. 1–4, pp. 357–368, 2015.

[27] H.-C. Liu, J.-X. You, X.-F. Ding, and Q. Su, "Improving risk evaluation in FMEA with a hybrid multiple criteria decision making method," *International Journal of Quality & Reliability Management*, vol. 32, no. 7, pp. 763–782, 2015.

[28] K.-H. Chang and C.-H. Cheng, "A novel general approach to evaluating the PCBA for components with different membership function," *Applied Soft Computing*, vol. 9, no. 3, pp. 1044–1056, 2009.

[29] M. Omidvar and F. Nirumand, "Risk assessment using FMEA method and on the basis of MCDM, fuzzy logic and grey theory: a case study of overhead cranes," *Journal of Health and Safety at Work*, vol. 7, no. 1, p. 63, 2017.

[30] S. H. Mirghafoori, M. R. Izadi, and A. Daei, "Analysis of the barriers affecting the quality of electronic services of libraries by VIKOR, FMEA and entropy combined approach in an intuitionistic-fuzzy environment," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 4, pp. 2441–2451, 2018.

[31] H. Zhao, J.-X. You, and H.-C. Liu, "Failure mode and effect analysis using MULTIMOORA method with continuous weighted entropy under interval-valued intuitionistic fuzzy environment," *Soft Computing*, vol. 21, no. 18, pp. 5355–5367, 2017.

[32] R. Fattahi and M. Khalilzadeh, "Risk evaluation using a novel hybrid method based on FMEA, extended MULTIMOORA, and AHP methods under fuzzy environment," *Safety Science*, vol. 102, pp. 290–300, 2018.

[33] P. Koomsap and T. Charoenchokdilok, "Improving risk assessment for customer-oriented FMEA," *Total Quality Management & Business Excellence*, vol. 29, no. 13-14, pp. 1563–1579, 2018.

[34] S. F. Liu, J. H. Cheng, Y. L. Lee, and F. R. Gau, "A case study on FMEA-based quality improvement of packaging designs in the TFT-LCD industry," *Total Quality Management & Business Excellence*, vol. 27, no. 3-4, pp. 413–431, 2016.

[35] S. K. Yang, C. Bian, X. Li, L. Tan, and D. X. Tang, "Optimized fault diagnosis based on FMEA-style CBR and BN for embedded software system," *International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3441–3453, 2018.

[36] E. Zarei, A. Azadeh, N. Khakzad, M. M. Aliabadi, and I. Mohammadfam, "Dynamic safety assessment of natural gas stations using Bayesian network," *Journal of Hazardous Materials*, vol. 321, pp. 830–840, 2017.

[37] B. H. Lee, "Using Bayes belief networks in industrial FMEA modeling and analysis," in *Proceedings of the Annual Reliability and Maintainability Symposium. 2001 Proceedings. International Symposium on Product Quality and Integrity*, pp. 7–15, Philadelphia, PA, USA, January 2001.

[38] L. Liu, D. Fan, Z. Wang et al., "Enhanced GO methodology to support failure mode, effects and criticality analysis," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1451–1468, 2019.

[39] H. Alyami, Z. Yang, R. Riahi, S. Bonsall, and J. Wang, "Advanced uncertainty modelling for container port risk analysis," *Accident Analysis & Prevention*, vol. 123, pp. 411–421, 2019.

[40] K.-H. Chang, Y.-C. Chang, and I.-T. Tsai, "Enhancing FMEA assessment by integrating grey relational analysis and the decision making trial and evaluation laboratory approach," *Engineering Failure Analysis*, vol. 31, pp. 211–224, 2013.

[41] K.-H. Chang, Y.-C. Chang, and Y.-T. Lee, "Integrating TOPSIS and DEMATEL methods to rank the risk of failure of FMEA," *International Journal of Information Technology & Decision Making*, vol. 13, no. 6, pp. 1229–1257, 2014.

[42] S. B. Tsai, J. Zhou, Y. Gao et al., "Combining FMEA with DEMATEL models to solve production process problems," *PLoS One*, vol. 12, no. 8, p. e0183634, 2017.

[43] M. Ekmekcioglu and A. C. Kutlu, "A fuzzy hybrid approach for fuzzy process FMEA: an application to a spindle manufacturing process," *International Journal of Computational Intelligence Systems*, vol. 5, no. 4, pp. 611–626, 2012.

[44] H. Safari, Z. Faraji, and S. Majidian, "Identifying and evaluating enterprise architecture risks using FMEA and fuzzy VIKOR," *Journal of Intelligent Manufacturing*, vol. 27, no. 2, pp. 475–486, 2016.

[45] S. Yousefi, A. Alizadeh, J. Hayati, and M. Baghery, "HSE risk prioritization using robust DEA-FMEA approach with undesirable outputs: a study of automotive parts industry in Iran," *Safety Science*, vol. 102, pp. 144–158, 2018.

[46] Y. M. Wang, K. S. Chin, G. K. K. Poon, and J. B. Yang, "Risk evaluation in failure mode and effects analysis using fuzzy weighted geometric mean," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1195–1207, 2009, in English.

[47] H.-C. Liu, J.-X. You, X.-J. Fan, and Q.-L. Lin, "Failure mode and effects analysis using D numbers and grey relational projection method," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4670–4679, 2014, in English.

[48] J. A. Rodger, P. Pankaj, and S. P. Gonzalez, "Decision making using a fuzzy induced linguistic ordered weighted averaging approach for evaluating risk in a supply chain," *International Journal of Advanced Manufacturing Technology*, vol. 70, no. 1–4, pp. 711–723, 2014.

[49] Q. Zhou and V. V. Thai, "Fuzzy and grey theories in failure mode and effect analysis for tanker equipment failure prediction," *Safety Science*, vol. 83, pp. 74–79, 2016.

[50] S. Mandal and J. Maiti, "Risk analysis using FMEA: fuzzy similarity value and possibility theory based approach," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3527–3537, 2014.

[51] N. Chanamool and T. Naenna, "Fuzzy FMEA application to improve decision-making process in an emergency department," *Applied Soft Computing*, vol. 43, pp. 441–453, 2016.

[52] X. Li, H. Li, B. Sun, and F. Wang, "Assessing information security risk for an evolving smart city based on fuzzy and grey FMEA," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 4, pp. 2491–2501, 2018.

[53] H.-C. Liu, P. Li, J.-X. You, and Y.-Z. Chen, "A novel approach for FMEA: combination of interval 2-tuple linguistic variables and gray relational analysis," *Quality and Reliability Engineering International*, vol. 31, no. 5, pp. 761–772, 2015.

[54] Z. Li and L. Chen, "A novel evidential FMEA method by integrating fuzzy belief structure and grey relational projection method," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 136–147, 2019.

[55] H.-C. Liu, J.-X. You, Q.-L. Lin, and H. Li, "Risk assessment in system FMEA combining fuzzy weighted average with fuzzy decision-making trial and evaluation laboratory," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 7, pp. 701–714, 2015.

[56] R. Meraj and S. N. Farhad, "Prediction of subsidence risk by FMEA using artificial neural network and fuzzy inference system," *International Journal of Mining Science and Technology*, vol. 25, no. 4, pp. 655–663, 2015.

[57] W. Y. Song, X. G. Ming, Z. Y. Wu, and B. T. Zhu, "A rough TOPSIS approach for failure mode and effects analysis in uncertain environments," *Quality and Reliability Engineering International*, vol. 30, no. 4, pp. 473–486, 2014, in English.

[58] H.-W. Lo, J. J. H. Liou, C.-N. Huang, and Y.-C. Chuang, "A novel failure mode and effect analysis model for machine tool risk analysis," *Reliability Engineering & System Safety*, vol. 183, pp. 173–183, 2019.

[59] J. Li, H. Fang, and W. Song, "Modified failure mode and effects analysis under uncertainty: a rough cloud theory-based approach," *Applied Soft Computing*, vol. 78, pp. 195–208, 2019.

[60] X. L. Zhou and Y. C. Tang, "Modeling and fusing the uncertainty of FMEA experts using an entropy-like measure with an application in fault evaluation of aircraft turbine rotor blades," *Entropy*, vol. 20, no. 11, p. 864, 2018.

[61] P. Chemweno, L. Pintelon, A.-M. De Meyer, P. N. Muchiri, A. Van Horenbeek, and J. Wakiru, "A dynamic risk assessment methodology for maintenance decision support," *Quality and Reliability Engineering International*, vol. 33, no. 3, pp. 551–564, 2017.

[62] J. He, T. Bao, J. Wu et al., "Reliability assessment and data processing techniques of the squib valve in pressurized water NPPs," *Nuclear Engineering and Design*, vol. 332, pp. 59–69, 2018.

[63] T. Jin and Y. Zhu, "First hitting time about solution for an uncertain fractional differential equation and application to an uncertain risk index model," *Chaos, Solitons & Fractals*, vol. 137, p. 109836, 2020.

[64] L. A. Zadeh, "A note on *Z*-numbers," *Information Sciences*, vol. 181, no. 14, pp. 2923–2932, 2011.

[65] K.-W. Shen and J.-Q. Wang, "Z-VIKOR method based on a new comprehensive weighted distance measure of *Z*-number and its application," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3232–3245, 2018.

[66] H.-G. Peng and J.-Q. Wang, "A multicriteria group decision-making method based on the normal cloud model with Zadeh's *Z*-numbers," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3246–3260, 2018.

[67] J.-Q. Wang, Y.-X. Cao, and H.-Y. Zhang, "Multi-criteria decision-making method based on distance measure and choquet integral for linguistic *Z*-numbers," *Cognitive Computation*, vol. 9, no. 6, pp. 827–842, 2017.

[68] J. Huang, D. Xu, H. Liu, and M. Song, "A new model for failure mode and effect analysis integrating linguistic *Z*-numbers and projection method," *IEEE Transactions on Fuzzy Systems*, p. 1, 2019.

[69] S. J. Ghoushchi, S. Yousefi, and M. Khazaeili, "An extended FMEA approach based on the Z-MOORA and fuzzy BWM for prioritization of failures," *Applied Soft Computing*, vol. 81, p. 105505, 2019.

Hindawi
Complexity
Volume 2021, Article ID 6614283, 17 pages
https://doi.org/10.1155/2021/6614283

Research Article

Big Archive-Assisted Ensemble of Many-Objective Evolutionary Algorithms

Wen Zhong,[1] Jian Xiong,[2] Anping Lin,[3] Lining Xing,[1] Feilong Chen,[4] and Yingwu Chen[1]

[1]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*School of Business Administration, Southwestern University of Finance and Economics, Chengdu, Sichuan 611130, China*
[3]*School of Electronic Information and Electrical Engineering, Xiangnan University, Chenzhou 423000, China*
[4]*Unit No. 78090, Chengdu 610031, China*

Correspondence should be addressed to Jian Xiong; xiongjian2017@swufe.edu.cn

Received 8 October 2020; Revised 11 November 2020; Accepted 28 January 2021; Published 18 February 2021

Academic Editor: Dan Selisteanu

Multiobjective evolutionary algorithms (MOEAs) have witnessed prosperity in solving many-objective optimization problems (MaOPs) over the past three decades. Unfortunately, no one single MOEA equipped with given parameter settings, mating-variation operator, and environmental selection mechanism is suitable for obtaining a set of solutions with excellent convergence and diversity for various types of MaOPs. The reality is that different MOEAs show great differences in handling certain types of MaOPs. Aiming at these characteristics, this paper proposes a flexible ensemble framework, namely, ASES, which is highly scalable for embedding any number of MOEAs to promote their advantages. To alleviate the undesirable phenomenon that some promising solutions are discarded during the evolution process, a big archive that number of contained solutions be far larger than population size is integrated into this ensemble framework to record large-scale nondominated solutions, and also an efficient maintenance strategy is developed to update the archive. Furthermore, the knowledge coming from updating archive is exploited to guide the evolutionary process for different MOEAs, allocating limited computational resources for efficient algorithms. A large number of numerical experimental studies demonstrated superior performance of the proposed ASES. Among 52 test instances, the ASES performs better than all the six baseline algorithms on at least half of the test instances with respect to both metrics hypervolume and inverted generational distance.

1. Introduction

In real-world applications, various kinds of optimization problems [1–3] have multiple conflicting objectives, such as design hybrid renewable energy systems [4], resource management for intelligent traffic [5, 6], optimization of wireless networks [7, 8], resource allocation for radar system [9, 10], and workflow scheduling in distributed environments [11]. These problems are often called multiobjective optimization problems (MOPs) and mathematically expressed as

$$\begin{cases} \text{Min} & F(\vec{x}) = \left[f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})\right] \\ \text{s.t.} & \vec{x} \in \Omega. \end{cases} \tag{1}$$

As shown in (1), a decision vector and its corresponding objective vector are denoted as $\vec{x}$ and $F(\vec{x})$. Then, the decision vector $\vec{x}$ can be detailed as $\vec{x} = (x_1, x_2, \ldots, x_n)$, where $x_i \in \vec{x}$ corresponds to the i-th decision variable and n represents the count of decision variables. For objective function, the $f_j(\vec{x}) \in F(\vec{x})$ is the j-th optimization objective, which maps each decision vector from n-dimensional decision space to a real number, i.e., $f_j(\vec{x})$: $R^n \longrightarrow R$. Generally speaking, MOPs with at least four optimization objectives (i.e., $m \geq 4$) are referred to many-objective optimization problems (MaOPs) [12–14].

On account of inherent conflicts among different optimization objectives, there is no one optimal solution that minimizes all the objective functions in unison and is replaced by a set of compromise solutions [15–17]. With regard to two feasible solutions $\vec{x}_1, \vec{x}_2 \in \Omega$, one solution $\vec{x}_1$ is said to be dominated by the other solution $\vec{x}_2$ (indicated as $\vec{x}_1 \succ \vec{x}_2$) iff $\vec{x}_2$ has lower or equal objective

value in each objective dimension than $\vec{x}_1$, and at least, one objective function of $\vec{x}_2$ has a lower value. Any feasible solution $\mathbf{x}^* \in \Omega$ that is not dominated by all other solutions is called the Pareto-optimal solution. For an MOP, the number of Pareto-optimal solutions is very large, even infinite. The solution set composing of all the Pareto-optimal solutions is referred to Pareto set (PS) in decision space, i.e., PS = $\left\{\vec{x}^* | \nexists \vec{x} \in \Omega, \vec{x} \prec \vec{x}^*\right\}$. The projection of PS onto objective space is referred to Pareto-optimal front (PF), i.e., PF = $\{F(\vec{x}) | \vec{x} \in \text{PS}\}$.

As population-based evolutionary algorithms can search and maintain a set of approximating solutions during each run, their advantages in dealing with MOPs have been demonstrated, and they have been used in various applications. Until now, a plethora of MOEAs have been developed and improved. On the basis of their environment selection mechanisms, these existing works generally fall into the following three categories: decomposition-based, indicator-based, and dominance relation-based MOEAs [18–20]. Decomposition-based MOEAs leverage a set of reference vectors to transform the original MOP into a series of subproblems, which concurrently are solved in a collaborative way [21–23]. For indicator-based MOEAs, instead of comparing solutions using objective vectors directly, they attempt to measure solutions by one single metric, such as hypervolume, and $R2$ [24, 25]. The MOEAs based on dominance relation, represented by NSGA-II [26], have been widely used, and there exist a large number of relevant works [18]. The environment selection mechanisms of MOEAs based on dominance relation consist of two main steps: (1) cluster the solutions into many groups using their dominance relations, such as Pareto dominance and fuzzy dominance [26, 27], and (2) a secondary metric, e.g., crowding distance and shift-based density estimation [28], is employed to sort the solutions in the last accepted group.

Unfortunately, no one single MOEA equipped with given parameter settings, mating and variation operators, and environmental selection mechanism is suitable for obtaining a set of solutions with excellent convergence and diversity for various types of MaOPs. The reality is that different MOEAs show great differences in handling certain types of MaOPs. The analyses in [29, 30] demonstrate that Pareto dominance-based selection approaches have competitive performance in maintaining good diversity, while decomposition-based and indicator-based selection approaches are able to provide strong convergence pressure.

To achieve better overall performance in solving a diverse range of MaOPs, up to present, there exist some works devoted to combining strengths of different MOEAs via ensemble of multiple mating operators, variation operators, or environmental selection strategies. For instance, the AdaBoost framework is employed to ensemble of multiple variation operators to promote search abilities on both the exploitation and exploration [30]. The work [31] makes use of two external archives for ensemble of dominance-based and indicator-based environmental selection approaches. Also, ensemble of mating selection strategies is studied [32]. Most of existing works are focused on ensemble of multiple approaches for a certain module, while few works consider integrating multiple complete MOEAs. However, each module of an MOEA has a significant impact on its overall performance, and even different approaches in one module are competitive in solving some specific types of MaOPs. Therefore, one of the emphases in this paper is to integrate and collaborate multiple complete MOEAs with quite different characteristics for solving MaOPs with various characteristics, such as nonlinearity, multimodality, and irregular PF shapes.

Solution selection for evolutionary multiobjective optimization is a tricky task. For two nondominated solutions $\vec{x}_1$ and $\vec{x}_2$, the following contradictions often appear in different evolutionary stages: $\vec{x}_1$ is better than $\vec{x}_2$ in some stages, while the fitness of $\vec{x}_2$ is better in other stages. The work [33] has provided a visual example to illustrate the above contradictions. These contradictions often lead to that some promising solutions are discarded during the evolution process. To avoid good solutions being discarded, Ishibuchi et al. suggested using unlimited external archives to store all the generated solutions [33, 34]. Nevertheless, maintaining an unlimited external archive during each generation inevitably results in unacceptable computational overhead. Therefore, another focus of this paper is how to effectively alleviate the negative phenomena that some promising solutions are discarded during the evolution process.

Aiming at the above two challenges, this paper mainly makes the following technical contributions:

A flexible ensemble framework is designed to integrate any number of MOEAs to promote their advantages for better solving a wide range of MOPs. In this framework, a big archive is embedded to store a large number of nondominated solutions to alleviate the undesirable phenomenon that some promising solutions are discarded during the evolution process.

We develop an efficient environmental selection strategy to maintain the big archive. The time complexity of this selection strategy is linear with the number of optimization objectives and the size of big archive.

An adaptive strategy is proposed to online reward competitive mating and variation operators to generate more offspring solutions, such combining their advantages for solving various MOPs.

Three representative algorithms (i.e., RVEA [15], VaEA [35], and SPEA2+SDE [28]) are selected to realize a prototype for the proposed framework. Also, the effectiveness of the proposal is verified by comparing it with six representative algorithms on 52 test instances, and the comparison results demonstrate the superiority of the proposal.

The structure of this paper is as follows. Section 2 reviews the related studies on MOEAs, especially on ensemble approaches. Then, the proposed approach is detailed in Section 3, followed by experimental verification in Section 4. At last, Section 5 concludes the paper and gives several research directions.

2. Related Work

A MOEA mainly contains the following three operators: mating operator, variation operator, and environment selection operator. During each generation, the first two operators are combined to generate a new offspring population, while environment selection operator is employed to select a population from the combination of offspring and parent populations. Over the past three decades, multiobjective evolutionary optimization community is booming, and a plethora of relevant approaches have been developed to solve MOPs and MaOPs.

2.1. Ensemble of Variation Operators. So far, there exist many works focusing on the variation operator, and the powerful evolutionary algorithms have been widely employed. To produce an offspring population, the simulated binary crossover [36] and the polynomial mutation [37] have been widely employed in multiobjective evolutionary optimization. Additionally, other evolutionary algorithms, such as differential evolution [38, 39], artificial bee colony [40], covariance matrix adaptation [41, 42], and particle swarm optimization [43], have been explored to reinforce search ability of multiobjective evolutionary optimization.

As we all know, different evolutionary algorithms possess quite different capabilities in terms of exploitation and exploration, and they have their own advantages in solving different types of MaOPs. To date, many efforts have been devoted to integrate multiple evolutionary algorithms. For instance, Venske et al. [44] suggested two mechanisms of strategy adaptation to integrate three versions of differential evolution, i.e., DE/rand/1/bin, DE/rand/2/bin, and DE/nonlinear, into decomposition-based evolutionary multiobjective optimization. Wang et al. [45] suggested a competitive and cooperative ensemble approach that simulated binary crossover and differential evolution are simultaneously used to generate new solutions in each generation, and the number of generated offspring solutions is assigned adaptively according to their average fitness improvement at generation. Santiago et al. [46] employed a fuzzy logic controller to dynamically choose variation operators during different evolutionary phases on the basis of variation operators' contributions in the past generations. Li et al. [47] made use of the sliding window to count the fitness contributions of multiple variation operators and also suggested a bandit-based adaptive operator selection mechanism to pick variation operators for decomposition-based MOEAs during one generation. Wang et al. [30] followed the AdaBoost ensemble framework in a machine learning area and then designed multiple subpopulation-assisted operator ensemble approach to adjust evolution times for different operators.

2.2. Ensemble of Environmental Selection Operators. From the perspective of environmental selection, the existing MOEAs can be roughly divided into three categories: decomposition-based [22, 48], dominance-based [26, 49], and indicator-based [25]. Many studies have revealed that there is no one environmental selection approach performing well on a wide range of scenarios. For instance, the decomposition-based and indicator-based MOEAs perform better on convergence pressure, while dominance-based MOEAs are capable of maintaining diversity. However, their respective shortcomings are also obvious. For decomposition-based MOEAs, their performance is heavily dependent on the PF shapes of MOPs [29]. The high complexity of indicator-based MOEAs often limits their scalability in the objective dimension [25]. Besides, dominance resistance [50] seriously hinders Pareto dominance-based MOEAs from solving MOPs with many objectives.

Driven by these facts, many efforts have been dedicated to integrating multiple environmental selection operators, aiming to combine their advantages. For instance, Li et al. [51] mixed decomposition-based and dominance-based environmental selection operators to make good trade-offs between convergence and diversity. The algorithms MOEA/D-M2M [52] and OPE-MOEA [53] combined decomposition-based and Pareto dominance-based strategies. In these two algorithms, a set of reference vectors were employed to partition the objective space, and the Pareto dominance-based strategy was used to maintain the solutions in each subspace. Zhang et al. [32] regarded multiple environmental selection approaches having different characteristics as voters. Then, the solutions elected by more voters are selected to construct the mating pool for generating new offspring population. Liang et al. [54] integrated indicator I_{ε^+} and dominance-based strategy to balance the convergence, diversity, and coverage for solving MaOPs.

3. Algorithm Design

In this section, we first design the flexible ensemble framework, followed by a prototype using three representative MOEAs. Also, the maintaining strategy with low time complexity for big archive is detailed.

3.1. Flexible Ensemble Framework. Most of the existing evolutionary multiobjective ensemble frameworks are dedicated to integrating multiple operators for a certain module, such as mating selection, offspring generation, and environment selection. Unlike them, this paper strives to design a flexible ensemble framework, which is scalable for embedding all modules of multiple MOEAs, because each module of an MOEA has a significant impact on its overall performance. Also, a big archive is added to alleviate the bad phenomenon that some promising solutions are discarded during the evolution process. The visual flow diagram of proposed ensemble framework is given in Figure 1.

In the proposed ensemble framework, the mating and variation operators of an existing MOEA are taken as a whole, which are denoted as MV. When K existing MOEAs are implemented, their mating and variation operators form a set, expressed as $\{MV1, MV2, \ldots, MVK\}$. Also, K populations, expressed as $\{P_1, P_2, \ldots, P_K\}$, are integrated into

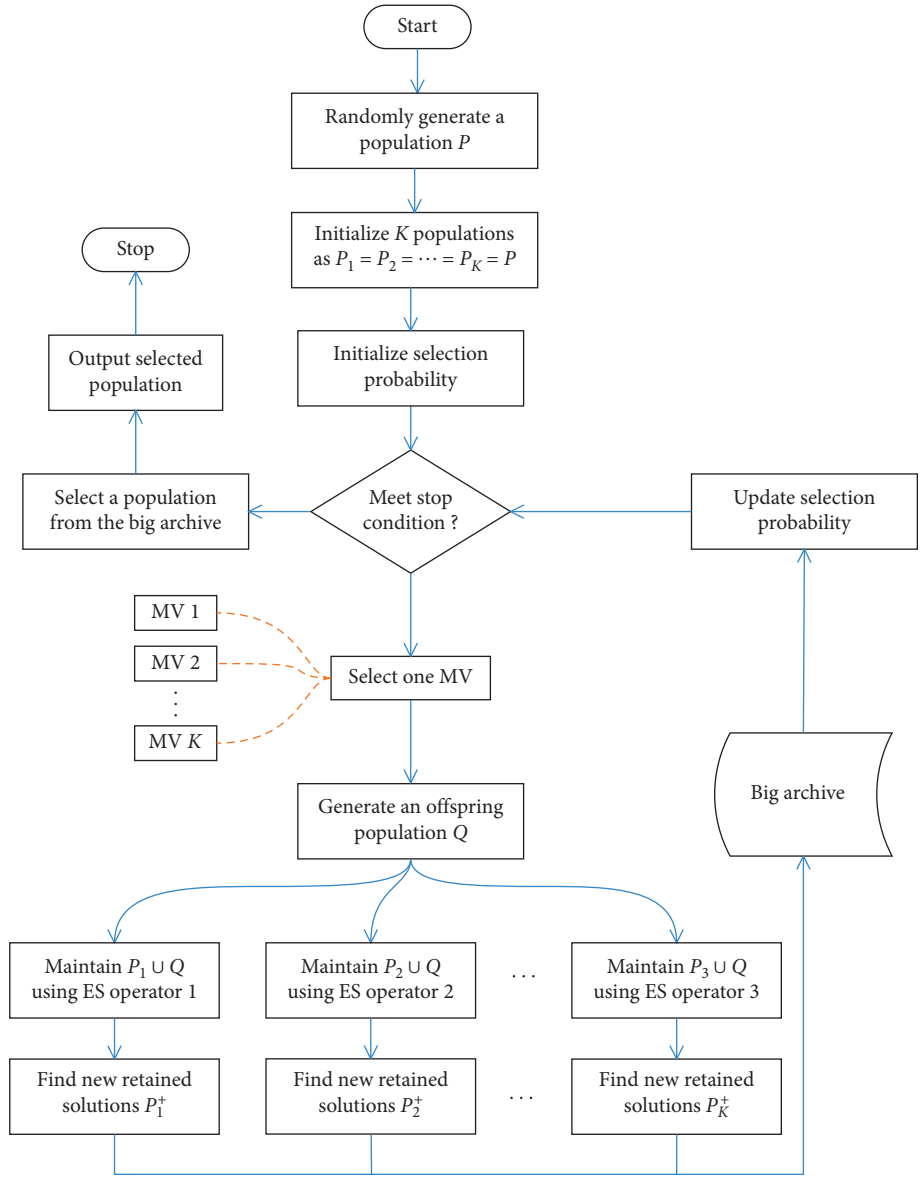


Figure 1: The flow diagram of the proposed flexible ensemble framework.

the framework in parallel, and each population is maintained by the environmental selection (ES) operator of the corresponding MOEA.

As illustrated in Figure 1, the K populations are first initialized with a randomly generated population, followed by initializing the selection probabilities for the K mating and variation operators. Then, the following six steps continue to loop until the stop condition is reached:

Step 1: according to selection probabilities, the roulette-wheel method [55] is employed to select one mating and variation operator.

Step 2: a new offspring population Q is generated by implementing selected mating and variation operator on the corresponding population. In other words, if the mating and variation operator of k-th MOEA is selected, the corresponding population refers to the k-th population.

Step 3: the K environment selection strategies run in parallel to update the K combined populations, i.e., $P_1 \cup Q, P_2 \cup Q, \ldots, P_K \cup Q$. In this ensemble framework, the newly generated population Q is combined with K populations, respectively.

Step 4: the new retained solutions from the Q by each environment selection strategy are selected, and the K sets of selected solutions are denoted as $P_1^+, P_2^+, \ldots, P_K^+$, respectively.

Step 5: add the K solution sets, $P_1^+, P_2^+, \ldots, P_K^+$, into the big archive, and update the big archive using a low-complexity maintenance mechanism, which will be detailed in the Section 3.2.2. At the same time, record the number of solutions retained in big archive for each environment selection operator.

Step 6: for an MOEA, the selection probability of its mating and variation operator is updated based on the

number of its selected solutions retained in the big archive.

Once the stopping condition is reached, an environmental selection operator will be applied to select an output population from the big archive.

Suppose the parameters, $r_1, r_2, \ldots, r_K$, respectively, represent the number of solutions retained in big archive from solution sets $P_1^+, P_2^+, \ldots, P_K^+$, and the selection probability p_k of the mating and variation operator from the k-th MOEA is updated as $p_k = r_k / \sum_{i=1}^{K} r_i$.

3.2. Prototype of the Proposed Framework. In this section, we implement three representative MOEAs and design a maintenance strategy to realize a prototype for the proposed ensemble framework.

3.2.1. Selected Algorithms. Three representative MOEAs, i.e., RVEA [15], VaEA [35], and SPEA2 + SDE [28], are selected and implemented. The brief descriptions of the above three algorithms and the reasons for selecting them are given as follows.

(1) RVEA. Its environmental selection operator employs a set of reference vectors to partition the objective spaces of MOPs into a series of nonoverlapping subspaces. For the solutions associated to the same subspace, the one with the smallest angle-penalized distance is selected.

The mating operator of RVEA randomly pairs solutions in the current population to construct mating pool, and then, the simulated binary crossover [36] and polynomial mutation [37] are employed to generate offspring population.

Similar to other decomposition-based MOEAs [22], the RVEA performs competitive on MOPs with PFs similar to the unit simplex, while its diversity deteriorates severely facing with MOPs with irregular PFs. Thus, we select algorithm RVEA as a representative for the branch of decomposition-based MOEAs.

(2) VaEA. The environmental selection operator of VaEA is a new version of Pareto dominance-based approach. It first makes use of nondominated sorting method to group the solutions into different levels. Then, for the solutions in a last accepted level, a maximum-minimum-angle strategy is designed to improve the diversity, while a worse-elimination strategy is employed to strengthen convergence.

The mating and variation operators of VaEA are the same as that of RVEA.

The VaEA belongs to the branch of Pareto dominance-based MOEAs and has lower time complexity. This makes it more popular in an ensemble framework.

(3) SPEA2 + SDE. A shift-based density estimation approach is designed to promote diversity maintenance in the environmental selection process. To measure the density of a solution $\vec{x}_1$, the density estimation approach shifts the objective values of other solutions. When the j-th objective value $f_j(\vec{x}_2)$ of solution $\vec{x}_2$ is lower than that of solution $\vec{x}_1$ in an objective, the $f_j(\vec{x}_2)$ is shifted to $f_j(\vec{x}_1)$.

On the basis of the solutions' fitness, the mating operator of SPEA2 + SDE employs tournament selection to construct the mating pool.

As demonstrated in work [56], the SPEA2 + SDE performs well on various MOPs compared with 21 MOEAs. Thus, we also choose it as a representative MOEA and implement it into the proposed ensemble framework.

Note that the simulated binary crossover and polynomial mutation are employed in original RVEA, VaEA, and SPEA2 + SDE as variation operators to generate offspring populations. To make the prototype have multiple variation operators, the variation operator of algorithm VaEA is replaced by the differential evolution [57].

As the VaEA is having parameter-less property and low time complexity compared with other two algorithms, when the stopping condition is reached, its environmental selection operator is applied to select an output population from the big archive.

3.2.2. Maintenance Mechanism for Big Archive. Using a big archive to store a large number of (e.g., ten times the population size) solutions is a good way to alleviate the negative phenomenon discarding some promising solutions during the evolution process. However, the time complexity of environmental selection operators in most existing algorithms is quadratic or even cubic in relation to the number of candidate solutions. For instance, time complexity of algorithms VaEA [35], NSGA-III [58], Two_Arch2 [31], and RVEA [15] is the square of the number of solutions, while that of algorithms GrEA [59] and SPEA2 + SDE [28] are cubic in relation to the number of solutions. Such high complexity limits the scalability of these environmental selection operators to maintain a big archive. Thus, a low-complexity maintenance mechanism is needed to maintain the big archive.

In this section, we design an efficient maintenance mechanism for maintaining the big archive. This mechanism first deletes all solutions that cannot dominate the Nadir point. Then, traversing each objective dimension, the interval consisting of minimum to maximum objective values is divided into a series of subintervals, and a solution with the smallest fitness is selected from each subinterval. The pseudocode of the proposed maintenance mechanism is shown in Algorithm 1.

As described in Algorithm 1, the inputs of the proposed maintenance mechanism are big archive, the nadir, and ideal points, while its output is the updated big archive. The set *De leA* is used to record the solutions whose objective vectors cannot dominate the nadir point $Z^{\max}$ and are initialized as empty (line 1). At first, all the solutions in the big archive are checked (line 2), and the solutions cannot dominate the nadir point are recorded in *De leA* (lines 3-4). After the solutions in *De leA* are removed (line 5), the remaining solutions will be selected from each objective dimension (line 7), and the set *SS* is employed to store the selected solutions. For an objective dimension, the interval consisting of minimum to maximum objective values is divided into H subintervals, and the solutions located in the

Input: A big archive $BigA$; Nadir point $Z^{\max}$; Ideal point $Z^{\min}$;
Output: An updated big archive A;
(1) $DeleA \longleftarrow \phi$
(2) **for each** $p \in BigA$ **do**
(3) **if** objective vector of p cannot dominate $Z^{\max}$ **then**
(4) $DeleA \longleftarrow DeleA \cup \{p\}$
(5) $BigA \longleftarrow BigA \setminus DeleA$
(6) $SS \longleftarrow \phi$
(7) **for** $j = 1 \longrightarrow m$**do**
(8) $S_1, S_2, \ldots, S_H \longleftarrow \phi$, where $H = \lceil N - |SS|/m - j + 1 \rceil$
(9) $Len \longleftarrow (N - |SS|/m - j + 1)$
(10) **foreach** $p \in BigA$ **do**
(11) $Ind \longleftarrow \lfloor f_i(p)/Len \rfloor$
(12) $S_{Ind} \longleftarrow S_{In\,d} \cup \{p\}$
(13) $S_{temp} \longleftarrow \phi$
(14) **for** $h = 1 \longrightarrow H$ **do**
(15) **if** S_h is not empty **then**
(16) $p \longleftarrow$ Select one solution with the smallest fitness from S_h
(17) $S_{temp} \longleftarrow S_{temp} \cup \{p\}$
(18) $BigA \longleftarrow BigA \setminus S_{temp}$
(19) $SS \longleftarrow SS \cup S_{temp}$

Algorithm 1: Maintenance mechanism.

h-th subinterval are recorded in the set S_h (lines 8–12). After that, one solution with the smallest fitness from each subinterval is selected (lines 14–17). For simplicity, the fitness of a solution is defined as the sum of all its objective values.

The time complexity of identifying the solutions that cannot dominate the nadir point is $O(m \cdot N)$ (lines 2–4), where m and N denote the number of objectives and solutions in the big archive. For an objective dimension, the proposed maintenance mechanism needs to check all the solutions to assign them into a corresponding subinterval, and thus, this process takes $O(N)$. The time complexity of selecting solutions from each subinterval is also $O(N)$. Then, the time complexity from lines 7 to 17 is $O(m \cdot (N + N)) = O(m \cdot N)$. In summary, the time complexity of the maintenance mechanism is $O(m \cdot N + m \cdot N) = O(m \cdot N)$.

Based on the above analysis, we can see that the time complexity of the proposed maintenance mechanism has a linear relationship with the number of objectives and the size of the big archive. This means that the maintenance mechanism has good scalability in both the objective dimension and solution scale and is suitable for dealing with optimization problems with many objectives and maintaining big archive.

4. Experimental Studies

This section carries out a large number of comparison experiments to verify the effectiveness of the proposal.

4.1. Experiment Design

4.1.1. Comparison Algorithms. To investigate the improvement of the proposed ensemble framework on the performance of existing MaOEAs, the three algorithms in its prototype, i.e., RVEA [15], VaEA [35], and SPEA2 + SDE [28], are first selected as comparison ones. In addition, three competitive algorithms, i.e., NSGA-III [58], MOEA/DD [51], and MOEA/D-AM2M [60], are chosen as comparison MaOEAs.

The brief descriptions of RVEA, VaEA, and SPEA2 + SDE are provided in Section 3.2. Here, we just give the brief descriptions for NSGA-III, MOEA/DD, and MOEA/D-AM2M as follows.

NSGA-III first employs the Pareto dominance relationships among solutions to classify them into many groups, and then, the solutions in the last accepted group are selected by a reference vector-based approach.

MOEA/DD improves the algorithm MOEA/D with a Pareto dominance strategy to strengthen the convergence for each subproblem.

MOEA/D-AM2M is a representative MOEA based on decomposition, and an adaptive mechanism is designed to adjust the reference vectors according to the distribution of obtained solutions. Also, a mechanism is employed to assign more search efforts for promising subproblems.

4.1.2. Benchmarks. The test functions MaF1–MaF13 [61] is the new versions of test suites DTLZ and WFG for assessing evolutionary many-objective optimization algorithms. These new test functions contain a variety of complicated features in real-world applications, such as degenerate, disconnected, biased deceptive, and mixed PF shapes, and complex Pareto sets. Based on these, the challenging test functions, i.e., MaF1–MaF13, are chosen to compare the proposed ASES with other six existing MaOEAs.

These test functions are scalable in the number of objectives, and a test function with a specific number of objectives is called as a test instance. For example, the MaF1 having 5 objectives is a test instance. To assess the scalability

of algorithm ASES in objective dimension, the number of objectives is set to 3, 6, 9, and 12, respectively. So, for 13 test functions with 4 settings for the number of objectives, there are 52 test instances comparing the seven algorithms.

To ensure fair comparisons, all MaOEAs are set to the same population size N for an instance. Besides, the population sizes of MaOEAs are set according to the objective number of test instances. Then, for the test instances with 3, 6, 9, and 12 objectives, the population sizes of the six comparison algorithms are set as 136, 182, 210, and 156, respectively.

For the proposed ASES, the size of output population is set as the population size of comparison algorithms. Besides, the size of big archive is set to ten times the size of output population.

4.1.3. Comparison Metrics. The metric inverted generational distance (IGD) [62] mainly tests the fitting degree of output population to the PF. When calculating the IGD, a set of uniformly distributed sample points P^* on the PF is required. In the experiment, the size of P^* is set to about 10000 for each test instance. With the sample points P^*, the IGD value of an output population P is calculated as follows:

$$\mathrm{IGD}(P) = \frac{\sum_{\overrightarrow{v} \in P^*} \min_{p \in P} \mathrm{Dis}(\overrightarrow{v}, F(p))}{|P^*|}, \quad (2)$$

where $\mathrm{Dis}(\overrightarrow{v}, F(p))$ denotes the distance between sample point $\overrightarrow{v}$ and objective vector $F(p)$ of solution p. According to formula (2), we can derive that a smaller *IGD* value of an output population means the better performance of the corresponding MaOEA.

The hypervolume (*HV*) [63] refers to the volume of a space consisting of a reference point and nondominated solutions in a population. This metric can simultaneously measure the diversity and convergence of the output population. A larger *HV* value indicates better performance of the corresponding algorithm. For each test instance, the reference point is set to 1.1 times of its nadir point in objective space. With a reference point $R = (r_1, r_2, \ldots, r_m)$, the *HV* value $\mathrm{HV}(P)$ of population P can be calculated as follows:

$$\mathrm{HV}(P) = \mathrm{Leb}\left(\bigcup_{p \in P} [f_1(p), r_1] \times \cdots \times [f_m(p), r_m] \right), \quad (3)$$

where $\mathrm{Leb}(\triangle)$ refers to the Lebesgue measure.

The metrics *IGD* and *HV* have been widely employed to assess the effectiveness of MaOEAs considering simultaneously both the diversity and convergence. The experiments also make use of them to compare our proposed ASES with other six existing MaOEAs.

4.1.4. Stop Condition. Similar to the works [15, 28, 35, 51, 58, 60], the stop condition is set as the maximal function evaluations for all the seven MaOEAs. For all the test instances, the number of maximal function evaluations is set as 100,000.

Regarding the variation operators simulated binary crossover (SBX), differential evolution (DE), and polynomial mutation, their parameters employ the default values provided by the platform PlatEMO (https://github.com/BIMK/PlatEMO). To be specific, the distribution index for SBX is set to 20; the crossover constant C and weighting factor F for DE are, respectively, set as 1.0 and 0.5; and we set $\eta = 20$ and $p_m = 1/n$ for the polynomial mutation.

4.2. Comparison Results and Analysis. In the experiments, the seven MaOEAs, i.e., RVEA, VaEA, SPEA2 + SDE, NSGA-III, MOEA/DD, MOEA/D-AM2M, and ASES, are repeated 31 times on all the test instances, and the mean and variance of *IGD* and *HV* values are reported in Tables 1 and 2. Besides, the Wilcoxon rank sum test is employed to carry out significance analysis between the proposed ASES and each comparison MaOEA, and the confidence level is set to 0.05. The symbols +, −, and ≈ in Tables 1 and 2 denote that the comparison MaOEA is better than, worse than, and similar to our proposal, respectively. For each test instance, the best metric value among the seven algorithms is highlighted with a gray background.

From Table 2, we can observe that, among 52 test instances, the proposed ASES has the smallest IGD values among seven MaOEAs on 27 test instances. These results demonstrate that the ASES performs better overall performance than other six comparison MaOEAs with respect to metric IGD. In detail, our proposal significantly outperforms RVEA, NSGA-III, VaEA, MOEA/DD, SPEA2 + SDE, and MOEA/D-AM2M on 46, 43, 41, 48, 39, and 49 out of the 52 test instances, respectively. By comparing the proposed ASES with the three comparison algorithms (i.e., RVEA, VaEA, and SPEA2 + SDE) in the prototype, we can derive that the proposed ensemble framework has the ability to further improve the performance of existing MaOEAs.

Facing test functions with irregular PF shapes, such as inverted MaF1, badly scaled MaF4, degenerate MaF6, disconnected MaF7, and complicated mixed MaF10, a part of reference vectors in decomposition-based MaOEAs are either not associated with any solution (e.g., RVEA) or associated with crowding solutions (e.g., NSGA-III, MOEA/DD, and MOEA/D-AM2M), resulting in their poor diversity. Since the proposal integrates Pareto dominance-based algorithms to make up for this deficiency, the proposed ASES performs better than these decomposition-based MaOEAs as a whole.

With respect to metric HV, the proposed ASES shows competitive advantages. As shown in Table 2, the proposal stands out as the strongest MaOEA, outperforming all the six comparison algorithms in 27 out of the 52 pairwise comparisons. These results again demonstrate that the ASES performs better overall performance than other six comparison MaOEAs in terms of metric HV. In detail, our proposal generates significantly larger HV values than RVEA, NSGA-III, VaEA, MOEA/DD, SPEA2 + SDE, and MOEA/D-AM2M on 47, 42, 47, 48, 36, and 44 out of the 52 test instances, respectively.

Table 1: Comparison of IGD values of the seven algorithms on solving 3-, 6-, 9-, and 12-objective test functions MaF1–MaF13.

MaOPs	m	RVEA	NSGA-III	VaEA	MOEA/DD	SPEA2 + SDE	MOEA/D-AM2M	ASES
MaF1	3	6.526e − 2 (4.0e − 3) −	4.837e − 2 (1.1e − 3) −	3.594e − 2 (3.2e − 4) −	5.424e − 2 (1.9e − 3) −	3.576e − 2 (5.8e − 4) −	5.997e − 2 (3.9e − 3) −	3.455e − 2 (1.9e − 4)
	6	3.373e − 1 (4.4e − 2) −	2.067e − 1 (1.5e − 2) −	1.482e − 1 (1.0e − 3) −	2.829e − 1 (5.8e − 3) −	1.453e − 1 (1.2e − 3) −	1.789e − 1 (6.0e − 3) −	1.429e − 1 (4.6e − 4)
	9	5.542e − 1 (6.2e − 2) −	2.691e − 1 (1.3e − 2) −	2.128e − 1 (1.2e − 3) ≈	4.318e − 1 (2.1e − 2) −	2.173e − 1 (1.6e − 3) −	3.190e − 1 (3.9e − 2) −	2.121e − 1 (7.3e − 4)
	12	6.196e − 1 (7.6e − 2) −	2.816e − 1 (4.6e − 3) −	2.590e − 1 (2.6e − 3) ≈	5.092e − 1 (4.2e − 2) −	2.864e − 1 (5.7e − 3) −	3.970e − 1 (4.6e − 2) −	2.600e − 1 (1.0e − 3)
MaF2	3	3.605e − 2 (1.3e − 3) −	2.915e − 2 (6.6e − 4) −	2.504e − 2 (2.4e − 4) −	3.981e − 2 (2.1e − 3) −	2.655e − 2 (4.0e − 4) −	3.037e − 2 (1.3e − 3) −	2.470e − 2 (1.7e − 4)
	6	1.540e − 1 (1.3e − 3) −	1.271e − 1 (3.0e − 3) −	1.230e − 1 (2.4e − 3) −	2.021e − 1 (7.9e − 2) −	1.086e − 1 (4.1e − 3) +	1.565e − 1 (8.1e − 3) −	1.165e − 1 (1.5e − 3)
	9	2.305e − 1 (7.0e − 3) −	2.198e − 1 (2.0e − 2) −	1.806e − 1 (2.9e − 3) −	2.349e − 1 (2.1e − 2) −	1.682e − 1 (3.9e − 3) +	2.218e − 1 (1.9e − 2) −	1.737e − 1 (2.6e − 3)
	12	5.235e − 1 (2.0e − 1) −	2.439e − 1 (4.6e − 2) −	2.136e − 1 (3.3e − 3) −	4.130e − 1 (5.3e − 2) −	2.057e − 1 (3.5e − 3) −	7.372e − 1 (8.7e − 2) −	2.000e − 1 (3.5e − 3)
MaF3	3	3.747e − 2 (2.5e − 3) +	3.897e − 2 (6.3e − 4) +	7.316e − 2 (2.8e − 2) −	4.509e − 2 (1.4e − 3) +	4.888e − 2 (3.6e − 3) ≈	4.121e − 1 (1.3e − 1) −	5.042e − 2 (2.2e − 3)
	6	1.286e − 1 (3.9e − 2) −	9.311e − 2 (7.8e − 3) ≈	6.605e + 0 (1.0e + 1) −	1.051e − 1 (6.7e − 3) −	9.726e − 2 (1.1e − − 2) ≈	4.876e − 1 (1.7e − 1) −	9.355e − 2 (1.0e − 2)
	9	3.706e − 1 (5.9e − − 1) −	7.091e + 3 (1.3e + 4) −	2.191e + 3 (2.7e + 3) −	3.484e + 0 (3.4e + 0) −	1.001e − 1 (5.3e − 3) −	5.304e − 1 (1.3e − 1) −	9.076e − 2 (5.0e − 3)
	12	9.248e − 2 (1.0e − 2) +	2.196e + 5 (5.9e + 5) −	8.381e + 3 (7.7e + 3) −	1.313e − 1 (1.3e − 1) −	1.004e − 1 (5.2e − 3) ≈	9.908e − 1 (1.6e + 0) −	1.157e − 1 (2.1e − 2)
MaF4	3	3.435e − 1 (1.1e − 1) −	2.756e − 1 (7.7e−3) −	3.112e − 1 (2.8e − 2) −	4.025e − 1 (2.7e−2) −	4.229e − 1 (5.4e − 2) −	1.240e + 0 (8.2e − 1) −	2.596e − 1 (7.5e − 3)
	6	9.678e + 0 (2.2e + 0) −	6.402e + 0 (8.7e − 1) −	5.013e + 0 (1.9e + 0) −	1.736e + 1 (5.6e − 1)−	7.610e + 0 (6.4e − 1) −	1.202e + 1 (2.5e + 0) −	3.879e + 0 (7.2e − 2)
	9	9.834e + 1 (3.1e + 1) −	4.810e + 1 (4.1e + 0) −	3.241e + 1 (2.2e + 0) −	1.816e + 2 (1.5e + 1) −	6.730e + 1 (8.8e + 0) −	1.401e + 2 (9.6e + 1) −	2.645e + 1 (5.3e − 1)
	12	9.228e + 2 (1.1e + 2) −	4.934e + 2 (3.1e + 1)−	2.432e + 2 (2.7e + 1) −	1.850e + 3 (4.1e + 1) −	7.004e + 2 (9.4e + 1) −	2.637e + 3 (3.1e + 3) −	2.061e + 2 (1.2e + 1)
MaF5	3	3.371e − 1 (4.1e−1) +	2.082e − 1 (1.0e − 4) +	2.160e − 1 (2.9e − 3) +	2.407e − 1 (1.8e − 3) +	4.364e − 1 (3.7e − 1) ≈	3.132e − 1 (1.7e − 2) +	7.527e − 1 (8.7e − 1)
	6	4.062e + 0 (2.1e − 2) −	4.074e + 0 (1.0e − 2) −	3.931e + 0 (7.4e − 2) ≈	1.631e + 1 (1.6e + 0) −	4.355e + 0 (2.0e − 1) −	4.277e + 0 (7.3e − 2) −	3.995e + 0 (4.6e − 2)
	9	5.260e + 1 (1.1e + 1) −	4.409e + 1 (6.2e − 1) −	2.817e + 1 (6.3e − 1) +	1.537e + 2 (3.1e + 0) −	3.457e + 1 (1.3e + 0) −	3.278e + 1 (1.9e + 0) −	2.978e + 1 (9.4e − 1)
	12	4.229e + 2 (4.9e + 1) −	4.391e + 2 (1.5e + 1) −	2.250e + 2 (1.3e + 1) +	1.081e + 3 (1.4e + 1) −	2.960e + 2 (1.5e + 1) −	2.369e + 2 (1.5e + 1) +	2.550e + 2 (7.9e + 0)

Table 1: Continued.

MaOPs	m	RVEA	NSGA-III	VaEA	MOEA/DD	SPEA2 + SDE	MOEA/D-AM2M	ASES
MaF6	3	2.951e − 2 (4.7e − 3) −	1.099e−2 (1.8e − 3) −	3.293e − 3 (4.1e − 5) −	2.604e − 2 (5.9e − 4) −	7.331e − 3 (6.4e − 4) −	9.156e − 2 (5.6e − 2) −	3.199e − 3 (1.1e − 5)
	6	7.088e − 2 (1.3e − 2) −	1.715e − 2 (7.4e − 3) −	3.162e − 3 (1.4e − 4) +	8.293e−2 (4.6e − 3) −	6.306e − 3 (7.7e − 4) −	2.345e − 1 (2.3e − 1) −	3.200e − 3 (1.2e − 4)
	9	8.847e − 2 (1.6e − 2) −	8.082e − 1 (1.6e + 0) −	7.034e − 1 (1.9e − 1) −	1.149e−1 (8.7e − 3) −	3.936e − 1 (2.7e−1) −	3.559e − 1 (2.2e − 1) −	2.576e−2 (7.1e − 2)
	12	2.201e − 1 (1.8e − 1) −	3.008e + 0 (7.0e + 0) −	5.025e − 1 (1.3e − 1) −	1.234e − 1 (5.9e − 3) −	4.640e − 1 (2.1e − 1) −	2.817e − 1 (2.3e − 1) −	2.052e − 2 (1.4e − 2)
MaF7	3	9.201e − 2 (3.1e − 3) −	6.095e − 2 (2.2e − 3) −	5.065e − 2 (1.5e − 3) −	2.537e − 1 (1.2e − 1) −	5.013e − 2 (2.1e − 3) −	9.078e − 2 (2.4e − 2) −	4.752e − 2 (4.1e − 4)
	6	6.365e − 1 (1.4e − 2) −	4.084e − 1 (1.9e − 2) −	4.338e − 1 (8.8e − 3) −	2.016e+0 (7.5e − 1) −	3.597e − 1 (3.7e − 2) +	3.975e − 1 (3.6e − 2) ≈	3.820e − 1 (9.3e − 3)
	9	1.759e + 0 (4.3e − 1) −	9.237e − 1 (6.6e − 2) −	7.876e − 1 (2.1e − 2) −	1.879e + 0 (5.5e − 1) −	8.929e − 1 (3.8e − 2) −	1.116e + 0 (2.7e − 1) −	7.193e − 1 (5.7e − 3)
	12	2.464e + 0 (4.0e − 1) −	1.517e + 0 (2.7e − 1) −	1.322e + 0 (2.3e − 2) −	3.038e + 0 (9.5e − 2) −	1.179e + 0 (1.6e − 2) ≈	5.918e + 0 (1.0e + 0) −	1.186e + 0 (1.0e − 2)
MaF8	3	1.141e − 1 (8.2e − 3) −	8.425e − 2 (3.6e − 3) −	5.749e − 2 (9.1e − 4) −	1.037e − 1 (5.3e − 3) −	6.719e − 2 (8.7e − 3) −	1.689e − 1 (2.9e − 2) −	5.352e − 2 (6.6e − 4)
	6	5.786e − 1 (9.3e − 2) −	2.377e − 1 (1.7e − 2) −	1.043e − 1 (2.1e − 3) −	4.302e − 1 (3.0e − 2)−	1.063e − 1 (5.6e − 3) −	3.469e − 1 (3.5e − 2) −	9.774e − 2 (1.5e − 3)
	9	8.866e − 1 (7.5e − 2) −	2.702e − 1 (5.4e − 2) −	1.250e − 1 (2.4e − 3) −	8.167e − 1 (2.3e − 2) −	1.293e − 1 (6.7e − 3) −	5.896e − 1 (1.0e − 1) −	1.187e − 1 (1.7e − 3)
	12	1.186e + 0 (1.1e − 1) −	3.689e − 1 (6.6e − 2) −	1.712e − 1 (2.6e − 3) −	1.099e + 0 (1.3e − 2) −	1.825e − 1 (1.9e − 2) −	6.908e − 1 (8.9e − 2) −	1.612e − 1 (2.8e − 3)
MaF9	3	5.085e − 2 (3.7e − 3) +	4.982e − 2 (7.1e − 4) +	3.753e − 1 (6.0e − 2) −	5.006e − 2 (8.3e − 4) +	5.261e − 2 (9.8e − 4) ≈	1.945e − 1 (3.0e − 2) −	6.667e − 2 (4.5e − 2)
	6	4.236e − 1 (1.0e − 1) −	5.697e − 1 (3.5e − 1) −	2.472e − 1 (9.1e − 2) −	3.513e − 1 (1.8e − 2) −	8.620e − 2 (7.5e − 4) +	5.902e − 1 (1.0e − 1) −	1.621e − 1 (2.1e − 2)
	9	8.163e − 1 (1.3e − 1) −	7.798e − 1 (1.1e + 0) −	3.623e − 1 (2.1e − 1) ≈	4.628e − 1 (6.1e − 2) −	1.067e − 1 (8.3e − 4) +	6.938e − 1 (1.4e − 1) −	2.433e − 1 (8.9e − 2)
	12	1.274e + 0 (2.7e − 1) −	2.512e + 0 (2.8e + 0) −	2.202e − 1 (2.3e − 2) ≈	7.173e − 1 (3.3e − 2) −	1.414e − 1 (9.7e − 4) +	2.212e + 0 (1.9e + 0) −	2.260e − 1 (2.4e − 2)
MaF10	3	1.904e − 1 (1.5e − 2) −	1.265e − 1 (6.8e − 3) −	1.450e − 1 (6.8e − 3) −	1.608e − 1 (2.8e − 2) −	1.582e − 1 (1.1e − 2) −	4.908e − 1 (1.4e − 1) −	1.229e − 1 (2.9e − 3)
	6	5.731e − 1 (1.8e − 2) −	6.788e − 1 (4.7e − 2) −	9.269e − 1 (9.6e − 2) −	1.139e + 0 (9.9e − 2) −	5.801e − 1 (1.8e − 2) −	1.203e + 0 (1.7e − 1) −	4.988e − 1 (2.1e − 3)
	9	8.858e − 1 (4.0e − 2) −	1.096e + 0 (5.4e − 2) −	1.779e + 0 (1.5e − 1) −	1.046e + 0 (5.4e − 2) −	9.305e − 1 (3.2e − 2) −	1.495e + 0 (1.1e − 1) −	8.179e − 1 (2.9e − 2)
	12	1.237e + 0 (6.7e − 2) −	1.359e + 0 (8.8e − 2) −	1.970e + 0 (1.7e − 1) −	1.456e + 0 (5.8e − 2) −	1.304e + 0 (4.5e − 2) −	1.758e + 0 (1.1e − 1) −	1.076e + 0 (8.5e − 3)

Table 1: Continued.

MaOPs	m	RVEA	NSGA-III	VaEA	MOEA/DD	SPEA2 + SDE	MOEA/D-AM2M	ASES
MaF11	3	1.548e − 1 (5.8e − 3) −	1.315e − 1 (6.4e − 4) +	1.394e − 1 (2.9e − 3) −	1.492e − 1 (1.4e − 3)−	1.768e − 1 (7.4e−3) −	1.943e − 1 (2.3e − 2) −	1.351e − 1 (2.2e − 3)
	6	5.514e − 1 (1.4e − 2) −	5.323e − 1 (2.7e − 3) +	5.533e − 1 (9.1e − 3) −	7.549e − 1 (7.9e − 3) −	6.632e − 1 (3.0e − 2) −	6.661e − 1 (2.9e − 2) −	5.364e − 1 (2.9e − 3)
	9	9.365e − 1 (3.5e − 2) −	1.107e + 0 (2.1e − 1) −	8.842e − 1 (1.6e − 2) −	1.179e + 0 (1.6e − 2) −	1.000e + 0 (2.4e − 2) −	1.019e + 0 (5.0e − 2) −	8.521e − 1 (1.2e − 2)
	12	1.314e + 0 (9.5e − 2) −	1.634e + 0 (1.4e − 1) −	1.231e + 0 (3.0e − 2) −	1.364e + 0 (2.2e − 2) −	1.316e + 0 (3.2e − 2) −	2.186e + 0 (5.1e − 1) −	1.133e + 0 (2.1e − 2)
MaF12	3	1.830e − 1 (1.1e − 3) −	1.807e − 1 (1.9e − 3) ≈	1.870e − 1 (3.3e − 3) −	1.936e − 1 (1.5e − 3) −	2.762e − 1 (1.0e − 2) −	3.091e − 1 (3.4e − 2) −	1.812e − 1 (1.6e − 3)
	6	1.534e + 0 (7.4e − 3) ≈	1.522e + 0 (4.9e − 3) +	1.543e + 0 (1.1e − 2) ≈	1.757e + 0 (2.5e − 2) −	1.744e + 0 (3.0e − 2) −	1.850e + 0 (9.1e − 2) −	1.534e + 0 (7.1e − 3)
	9	3.782e + 0 (3.6e − 2) −	3.892e+0 (1.3e − 1) −	3.555e + 0 (2.8e − 2) −	5.084e + 0 (3.6e − 1) −	3.856e + 0 (3.3e − 2) −	4.051e + 0 (1.5e − 1) −	3.549e + + 0 (1.0e − 2)
	12	5.978e + 0 (1.4e − 1) ≈	6.553e + 0 (2.1e − 1) −	5.798e + 0 (2.2e − 2) +	7.219e + 0 (1.6e − 1) −	6.808e + 0 (2.3e − 1) −	6.563e + 0 (5.5e − 1) −	5.874e + 0 (3.3e − 2)
MaF13	3	7.662e − 2 (6.3e − 3) −	6.644e − 2 (6.9e − 3) ≈	8.519e − 2 (1.5e−2) −	5.349e − 2 (2.7e − 3) +	8.026e − 2 (8.8e − 3) −	2.234e − 1 (7.7e − 2) −	5.999e − 2 (6.5e − 3)
	6	6.993e − 1 (1.1e − 1) −	2.131e − 1 (2.6e − 2) −	1.633e − 1 (3.0e − 2) −	2.286e − 1 (1.6e − 2) −	9.811e − 2 (9.9e − 3) −	3.143e − 1 (3.1e − 2) −	8.623e − 2 (7.8e − 3)
	9	7.783e − 1 (2.5e − 1) −	2.400e − 1 (3.3e − 2) −	1.576e − 1 (2.8e − 2) −	3.742e − 1 (3.8e − 2) −	1.154e − 1 (2.3e − 2) −	3.151e − 1 (5.1e − 2) −	9.630e − 2 (1.1e − 2)
	12	1.128e + 0 (4.2e − 1) −	2.895e − 1 (1.8e − 2) −	2.084e − 1 (2.0e − 2) −	3.921e − 1 (4.8e − 2) −	1.476e − 1 (3.0e − 2) ≈	4.147e − 1 (8.4e − 2) −	1.258e − 1 (2.1e − 2)
+/−/ ≈		4/46/2	6/43/3	5/41/6	4/48/0	6/39/7	2/49/1	

The parallel coordinate [64] is a technique for displaying high-dimensional vectors in a two-dimensional graph, where each dimension of high-dimensional vectors is plotted on a vertical axis in the two-dimensional graph, and then, a high-dimensional vector is presented by a polyline connecting points on each axis. It has been frequently employed to visualize the distributions of output populations in evolutionary many-objective optimization community.

To graphically illustrate both the convergence and diversity of the seven algorithms, the parallel coordinate is employed to plot the distributions of their output populations corresponding to the smallest IGD values among 30 runs on 9-objective MaF4, MaF6, and MaF10, as illustrated in Figures 2–4.

The PF shape of test function MaF4 is inverted and badly scaled, and also, the number of local Pareto-optimal fronts in this function is up to $(3^{n+1-m} - 1)$. For the badly scaled PF of MaF4, as shown in Figure 2(a), the interval of the j-dimension objective values is from 0 to 2^j. From Figures 2(b) and 2(g), we can see that some solutions in the output populations of RVEA and MOEA/D-AM2M are far from convergent. Compared with the above two MaOEAs, the convergences of algorithms NSGA-III and VaEA are much better, but there exists still a part of solutions that fail to converge to dominate the nadir point of PF. For algorithm MOEA/DD, as shown in Figure 2(e), although it has good convergence, its diversity is poor. For instance, the values of the output population in 9-th objective only covers the interval [50, 260], while the interval covered by PF in this objective is [0, 512]. Comparing Figures 2(h) and 2(a), we can find out that the output population of algorithm ASES can fit the true PF very well. These visual results can explain that the proposal has better performance than the five comparison algorithms RVEA, NSGA-III, VaEA, MOEA/DD, and MOEA/D-AM2M with respect to both the IGD and HV, which are shown in Tables 1 and 2. Intuitively, SPEA2 + SDE shows good convergence and diversity, as shown in Figure 2(f). According to the IGD and HV values of algorithms SPEA2 + SDE and ASES on 9-objective MaF4,

Table 2: Comparison of HV values of the seven algorithms on solving 3-, 6-, 9-, and 12-objective test functions MaF1–MaF13.

MaOPs	m	RVEA	NSGA-III	VaEA	MOEA/DD	SPEA2 + SDE	MOEA/D-AM2M	ASES
MaF1	3	1.921e − 1 (5.0e − 3) −	2.146e − 1 (5.9e−4) −	2.236e − 1 (5.8e − 4) −	2.107e − 1 (1.8e − 3) −	2.262e − 1 (4.5e − 4) −	2.071e − 1 (3.0e − 3) −	2.277e − 1 (2.4e − 4)
	6	3.467e − 4 (1.5e − 4) −	8.496e − 4 (1.0e − 4) −	1.640e − 3 (4.8e − 5) −	5.475e − 4 (1.8e − 5) −	1.905e − 3 (9.0e − 5) ≈	1.546e − 3 (9.1e − 5) −	1.938e − 3 (5.3e − 5)
	9	1.180e − 7 (6.2e − 8) −	1.872e − 6 (4.1e − 7) −	3.415e − 6 (1.9e − 6) ≈	4.398e − 7 (1.3e − 7) −	4.139e − 6 (1.3e − 6) +	1.960e − 6 (7.9e − 7) −	3.300e − 6 (2.3e − 6)
	12	4.714e − 11 (2e − 11) +	4.050e − 9 (5e − 10) +	0.000e + 0 (0.0e + 0) ≈	2.472e − 10 (1e − 10) +	0.000e + 0 (0.0e + 0) ≈	1.078e − 9 (4e − 10) +	0.000e + 0 (0.0e + 0)
MaF2	3	2.343e − 1 (1.4e − 3) −	2.424e − 1 (8.9e − 4) −	2.456e − 1 (1.1e − 3) −	2.283e − 1 (2.0e − 3) −	2.482e − 1 (5.9e − 4) −	2.465e − 1 (1.3e − 3) −	2.483e − 1 (4.9e − 4)
	6	2.079e − 1 (3.0e − 3) −	2.235e − 1 (2.8e − 3) −	2.389e − 1 (2.8e − 3) −	1.819e − 1 (1.7e − 2) −	2.516e − 1 (2.4e − 3) −	2.339e − 1 (2.3e − 3) −	2.525e − 1 (1.1e − 3)
	9	1.693e − 1 (4.0e − 3) −	2.095e − 1 (6.6e − 3) −	2.190e − 1 (1.8e − 3) −	1.831e − 1 (3.4e − 3) −	2.264e − 1 (1.9e − 3) −	1.970e − 1 (8.4e − 3) −	2.312e − 1 (1.7e − 3)
	12	7.143e − 2 (1.7e − 2) −	1.703e − 1 (1.1e − 2) −	1.987e − 1 (3.8e − 3) −	1.014e − 1 (3.4e − 3) −	2.119e − 1 (2.7e − 3) −	1.177e − 1 (4.7e − 2) −	2.264e − 1 (2.5e − 3)
MaF3	3	9.601e − 1 (2.0e − 3) ≈	9.609e − 1 (6.9e − 4) +	9.408e − 1 (1.7e − 2) −	9.566e − 1 (1.0e − 3) −	9.541e − 1 (1.4e − 3) −	5.633e − 1 (2.0e − 1) −	9.592e − 1 (8.3e − 4)
	6	9.948e − 1 (9.7e − 3) −	9.996e − 1 (3.5e − 4) ≈	5.486e − 1 (4.7e − 1) −	9.941e − 1 (2.4e − 3) −	9.932e − 1 (2.4e − 3) −	7.115e − 1 (2.3e − 1) −	9.996e − 1 (5.6e − 5)
	9	8.173e − 1 (3.8e − 1) −	1.030e − 1 (3.1e − 1) −	0.000e + 0 (0.0e + 0) −	2.642e − 1 (4.1e − 1) −	9.971e − 1 (1.1e − 3) −	7.597e − 1 (2.2e − 1) −	9.999e − 1 (6.0e − 5)
	12	9.990e − 1 (1.3e − 3) +	0.000e + 0 (0.0e + 0) −	0.000e + 0 (0.0e + 0) −	9.449e − 1 (1.6e − 1) −	9.980e − 1 (5.1e − 4) ≈	6.184e − 1 (2.7e − 1) −	9.967e − 1 (7.7e − 3)
MaF4	3	5.174e − 1 (2.7e − 2) −	5.295e − 1 (3.3e − 3) −	5.322e − 1 (6.8e − 3) −	5.245e − 1 (4.0e − 3) −	5.346e − 1 (2.9e − 3) −	3.909e − 1 (9.0e − 2) −	5.404e − 1 (3.0e − 3)
	6	1.712e − 3 (8.2e − 4) −	1.172e − 2 (4.6e − 3) −	2.815e − 2 (1.0e − 2) −	2.294e − 3 (7.7e − 4) −	1.622e − 2 (3.8e − 3) −	6.393e − 3 (3.9e − 3) −	4.042e − 2 (1.7e − 3)
	9	1.716e − 6 (1.6e − 6) −	7.101e − 4 (1.0e − 4) −	3.562e − 4 (8.1e − 5) −	1.334e − 6 (3.4e − 7) −	2.425e − 5 (1.1e − 5) −	1.226e − 5 (1.6e − 5) −	8.429e − 4 (6.4e − 5)
	12	2.89e − 10 (1e − 10) −	1.051e − 5 (1.4e − 6) ≈	4.163e − 6 (1.6e − 6) −	5.03e − 10 (1e − 10) −	1.378e − 8 (7.0e − 9) −	3.560e − 9 (9.3e − 9) −	9.185e − 6 (1.9e − 6)
MaF5	3	5.459e − 1 (7.1e − 2) +	5.684e − 1 (8.0e − 5) +	5.634e − 1 (5.4e−4) ≈	5.559e − 1 (8.8e − 4) +	5.426e − 1 (6.8e−2) +	5.566e − 1 (1.9e − 3) ≈	5.020e−1 (1.0e − 1)
	6	8.545e − 1 (1.9e − 2) −	8.602e − 1 (4.4e − 4) ≈	8.382e − 1 (2.7e − 3) −	6.021e − 1 (4.8e − 2) −	8.245e − 1 (4.5e − 3) −	8.650e − 1 (2.0e − 3) +	8.601e − 1 (1.9e − 3)
	9	9.302e − 1 (3.0e − 3) −	9.511e − 1 (2.7e − 4) +	9.148e − 1 (9.3e − 3) −	5.775e − 1 (1.8e − 2) −	8.697e − 1 (6.0e − 3) −	9.536e − 1 (1.0e − 3) +	9.463e − 1 (1.4e − 3)
	12	8.864e − 1 (2.6e − 2) −	9.721e − 1 (2.9e − 4) +	9.443e − 1 (5.0e − 3) −	4.846e − 1 (6.1e − 2) −	8.610e − 1 (1.9e − 2) −	9.724e − 1 (3.1e − 3) +	9.621e − 1 (1.6e − 3)
MaF6	3	1.821e − 1 (3.4e − 3) −	1.961e − 1 (7.5e − 4) −	2.007e − 1 (5.4e − 5) −	1.861e − 1 (3.5e − 4) −	2.004e − 1 (1.1e − 4) −	1.604e − 1 (1.9e − 2) −	2.008e − 1 (1.5e − 5)
	6	1.045e − 1 (2.9e − 3) −	1.152e − 1 (1.9e − 3) −	1.182e − 1 (3.0e − 4) ≈	1.028e − 1 (1.2e − 3) −	1.173e − 1 (4.1e − 4) −	1.065e − 1 (6.2e − 3) −	1.182e − 1 (2.9e − 4)
	9	9.520e − 2 (1.2e − 3) −	3.237e − 2 (4.1e − 2) −	2.542e − 4 (8.0e − 4) −	9.495e − 2 (8.6e − 4) −	3.068e − 2 (4.9e − 2) −	9.815e − 2 (2.1e − 3) −	1.023e − 1 (2.6e − 3)
	12	9.169e − 2 (8.1e − 4) −	2.917e − 2 (3.0e − 2) −	1.448e − 16 (4e − 16) −	9.334e − 2 (8.8e − 4) −	1.203e − 2 (2.2e − 2) −	9.450e − 2 (1.3e − 3) −	9.521e − 2 (4.7e − 3)
MaF7	3	2.630e − 1 (3.3e − 3) −	2.766e − 1 (8.3e − 4) −	2.804e − 1 (5.8e − 4) −	2.304e − 1 (1.2e − 2) −	2.805e − 1 (4.8e − 4) −	2.728e − 1 (3.7e − 3) −	2.827e − 1 (1.8e − 4)
	6	1.932e − 1 (6.5e − 3) −	2.186e − 1 (4.7e − 3) −	2.148e − 1 (5.2e − 3) −	8.714e − 2 (1.3e − 3) −	2.348e − 1 (8.9e − 3) −	2.395e − 1 (5.4e − 3) −	2.509e − 1 (1.6e − 3)
	9	1.340e − 1 (2.0e − 2) −	1.616e−1 (1.2e − 2) −	1.450e − 1 (7.4e − 3) −	4.499e − 3 (1.3e − 2) −	3.567e − 2 (7.7e − 3) −	1.981e − 1 (8.5e − 3) +	1.906e − 1 (4.7e − 3)
	12	1.151e − 1 (5.1e−2) −	5.941e − 2 (2.8e − 2) −	1.222e − 1 (5.2e − 3) −	1.064e−5 (5.2e − 6) −	1.980e − 2 (2.1e−2) −	1.294e − 1 (1.3e − 2) ≈	1.337e−1 (6.4e−3)

Table 2: Continued.

MaOPs	m	RVEA	NSGA-III	VaEA	MOEA/DD	SPEA2 + SDE	MOEA/D-AM2M	ASES
MaF8	3	2.513e − 1 (4.1e − 3) −	2.704e − 1 (1.5e−3) −	2.818e−1 (4.6e − 4) −	2.586e − 1 (3.2e − 3) −	2.754e − 1 (3.7e − 3) −	2.274e − 1 (1.4e − 2) −	2.837e − 1 (3.8e − 4)
	6	5.047e − 2 (5.5e − 3) −	6.971e − 2 (3.8e − 3) −	9.065e − 2 (2.2e − 4) −	5.008e − 2 (1.6e − 3) −	9.113e − 2 (3.5e − 4) ≈	6.307e − 2 (3.6e − 3) −	9.128e − 2 (3.6e − 4)
	9	6.568e − 3 (9.1e−4) −	1.371e − 2 (1.0e − 3) −	1.694e−2 (1.3e − 4) −	9.741e − 3 (3.5e−4) −	1.718e − 2 (1.1e − 4) +	9.492e − 3 (1.0e − 3) −	1.705e − 2 (1.4e − 4)
	12	7.638e − 4 (2.0e − 4) −	2.647e − 3 (1.3e − 4) −	3.322e − 3 (4.5e − 5) −	1.861e − 3 (4.0e − 5) −	3.406e − 3 (4.8e − 5) +	1.682e − 3 (2.3e − 4) −	3.338e − 3 (6.4e − 5)
MaF9	3	8.427e − 1 (3.3e − 3) +	8.441e − 1 (6.1e − 4) +	6.245e − 1 (2.3e−2) −	8.437e − 1 (6.4e − 4) +	8.324e − 1 (2.4e − 3) +	7.320e − 1 (2.3e − 2) −	8.321e − 1 (3.0e−2)
	6	7.209e − 2 (1.3e − 2) −	6.900e − 2 (2.6e − 2) −	1.018e−1 (1.5e − 2) −	7.671e − 2 (2.3e − 3) −	1.358e−1 (2.6e − 4) +	5.688e − 2 (8.6e − 3) −	1.197e − 1 (5.0e − 3)
	9	1.128e − 2 (2.1e − 3) −	2.081e − 2 (8.4e − 3) −	2.526e − 2 (9.6e − 3) −	1.886e − 2 (1.1e − 3) −	3.829e − 2 (2.3e − 4) +	1.326e − 2 (4.0e − 3) −	3.005e − 2 (4.7e − 3)
	12	1.070e−3 (3.9e − 4) −	1.204e − 3 (1.0e−3) −	4.776e − 3 (2.7e − 4) ≈	2.791e − 3 (2.0e−4) −	5.804e − 3 (8.3e − 5) +	6.634e − 4 (5.7e − 4) −	4.640e−3 (3.2e − 4)
MaF10	3	9.214e − 1 (1.3e − 2) −	9.464e − 1 (1.0e−3) −	9.406e − 1 (1.8e − 3) −	9.319e − 1 (1.5e − 2) −	9.395e − 1 (1.6e − 3) −	8.328e − 1 (5.2e − 2) −	9.473e − 1 (1.3e − 3)
	6	9.050e − 1 (2.3e − 2) −	8.496e − 1 (3.5e − 2) −	6.860e − 1 (3.7e − 2) −	6.169e − 1 (5.3e − 2) −	9.961e − 1 (4.1e − 4) −	8.439e − 1 (4.3e − 2) −	9.989e − 1 (3.4e − 4)
	9	9.576e−1 (5.4e − 2) −	8.106e − 1 (4.3e−2) −	4.752e − 1 (5.2e − 2) −	8.806e−1 (7.0e − 2) −	9.969e − 1 (5.3e−4) −	8.494e − 1 (6.7e − 2) −	9.990e − 1 (6.0e−4)
	12	9.974e − 1 (7.0e − 4) −	9.982e − 1 (7.1e − 4) −	5.195e − 1 (5.9e−2) −	9.259e − 1 (7.3e − 2) −	9.970e − 1 (4.6e − 4) −	8.795e − 1 (5.4e − 2) −	9.998e − 1 (7.0e − 5)
MaF11	3	9.300e − 1 (9.6e − 4) −	9.359e − 1 (5.7e − 4) +	9.297e − 1 (6.7e − 4) −	9.323e − 1 (1.9e − 3) −	9.331e − 1 (2.1e − 3) ≈	9.365e − 1 (9.2e − 4) +	9.347e−1 (1.1e − 3)
	6	9.847e − 1 (5.7e − 3) −	9.944e − 1 (9.5e − 4) −	9.908e − 1 (1.2e − 3) −	9.686e − 1 (6.2e − 3) −	9.849e − 1 (3.5e − 3) −	9.955e − 1 (7.2e − 4) −	9.984e − 1 (3.5e − 4)
	9	9.827e − 1 (2.9e−3) −	9.948e − − 1 (3.2e − 3) −	9.898e − 1 (2.9e−3) −	9.559e − 1 (7.4e − 3) −	9.900e − 1 (1.8e − 3) −	9.936e − 1 (1.9e − 3) −	9.981e − 1 (9.0e − 4)
	12	9.660e − 1 (6.9e − 3) −	9.963e − 1 (1.5e − 3) −	9.927e − 1 (1.3e − 3) −	9.651e − 1 (1.0e − 2) −	9.897e − 1 (1.0e − 3) −	9.752e − 1 (1.0e − 2) −	9.981e − 1 (8.4e − 4)
MaF12	3	5.456e − 1 (1.8e − 3) −	5.448e − 1 (2.2e − 3) −	5.387e − 1 (4.1e − 3) −	5.372e − 1 (2.6e − 3) −	5.502e − 1 (3.2e − 3) −	5.337e − 1 (5.7e − 3) −	5.531e − 1 (2.8e − 3)
	6	7.873e − 1 (9.7e − 3) −	7.865e − 1 (7.5e − 3) −	7.611e − 1 (1.0e − 2) −	7.154e − 1 (3.5e − 2) −	7.838e − 1 (4.4e − 3) −	7.792e − 1 (1.0e − 2) −	8.039e − 1 (3.9e − 3)
	9	8.309e − 1 (4.2e − 2) −	8.522e − 1 (2.1e − 2) −	7.614e − 1 (6.9e − 2) −	6.537e − 1 (5.0e − 2) −	8.317e − 1 (9.4e − 3) −	8.249e − 1 (8.7e − 3) −	8.619e − 1 (1.1e − 2)
	12	7.954e − 1 (4.1e − 2) −	8.032e − 1 (6.4e − 2) −	7.868e − 1 (6.0e − 2) −	6.701e − 1 (5.5e − 2) −	8.165e − 1 (7.0e − 2) −	7.759e − 1 (1.9e − 2) −	8.561e − 1 (5.1e − 2)
MaF13	3	5.092e − 1 (9.4e − 3) −	5.238e − 1 (1.2e − 2) −	4.903e − 1 (2.2e − 2) −	5.452e − 1 (6.1e − 3) ≈	5.561e − 1 (7.6e − 3) +	4.195e − 1 (3.7e − 2) −	5.429e − 1 (9.7e − 3)
	6	1.274e − 1 (1.4e − 2) −	1.801e − 1 (1.3e − 2) −	1.877e − 1 (2.2e − 2) −	1.193e − 1 (2.9e − 2) −	2.362e − 1 (5.0e − 3) +	1.163e − 1 (4.6e − 2) −	2.342e − 1 (6.1e − 3)
	9	9.345e−2 (1.0e − 2) −	1.206e − 1 (9.1e − 3) −	1.220e − 1 (1.6e − 2) −	6.044e − 2 (1.8e − 2) −	1.561e − 1 (3.4e − 3) +	9.009e − 2 (3.7e − 2) −	1.507e − 1 (5.6e − 3)
	12	6.608e − 2 (2.5e − 2) −	9.007e − 2 (8.6e − 3) −	9.123e − 2 (9.2e − 3) −	2.836e − 2 (1.2e − 2) −	1.139e − 1 (1.7e − 3) −	2.327e − 2 (2.6e − 2) −	1.157e − 1 (1.2e − 2)
+/−/ ≈		4/47/1	7/42/3	0/47/5	3/48/1	10/36/6	7/44/1	

we can derive that the proposed ASES has better convergence and diversity than the comparison algorithm SPEA2 + SDE.

The test function MaF6 is a representative of MaOPs with degenerate PFs. To visually compare the capability of the seven MaOEAs in dealing with degenerate PFs, their output populations corresponding to the smallest IGD values among 30 runs on 9-objective MaF6 are plotted in parallel coordinate, as illustrated in Figure 3. For test function MaF6 with 9 objectives, the ninth objective on the PF ranges from 0 to 1.0, while the values of the eighth objective on the PF are between 0 and 0.7, as shown in the true PF in Figure 3(a). From Figures 3(b) and 3(d)–3(g), it is obvious that most of the solutions in output populations

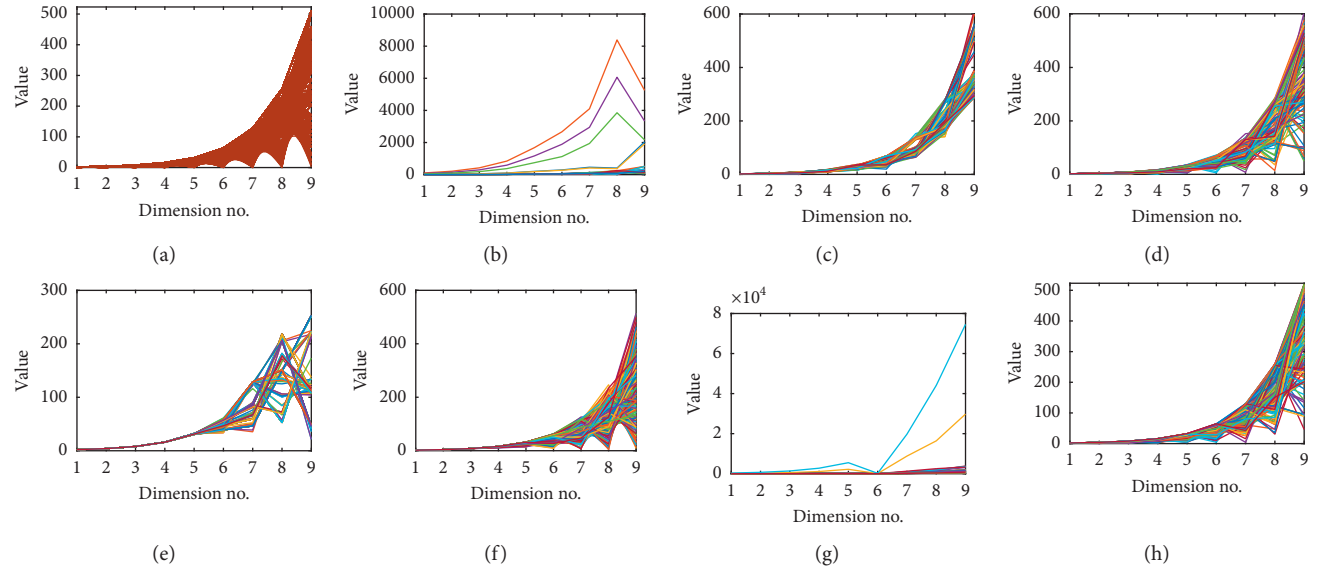


Figure 2: For 9-objective MaF4, its true PF and output populations of seven algorithms in parallel coordinate: (a) true PF of MaF4; (b) RVEA on MaF4; (c) NSGA-III on MaF4; (d) VaEA on MaF4; (e) MOEA/DD on MaF4; (f) SPEA2 + SDE on MaF4; (g) MOEA/D-AM2M on MaF4; (h) ASES on MaF4.

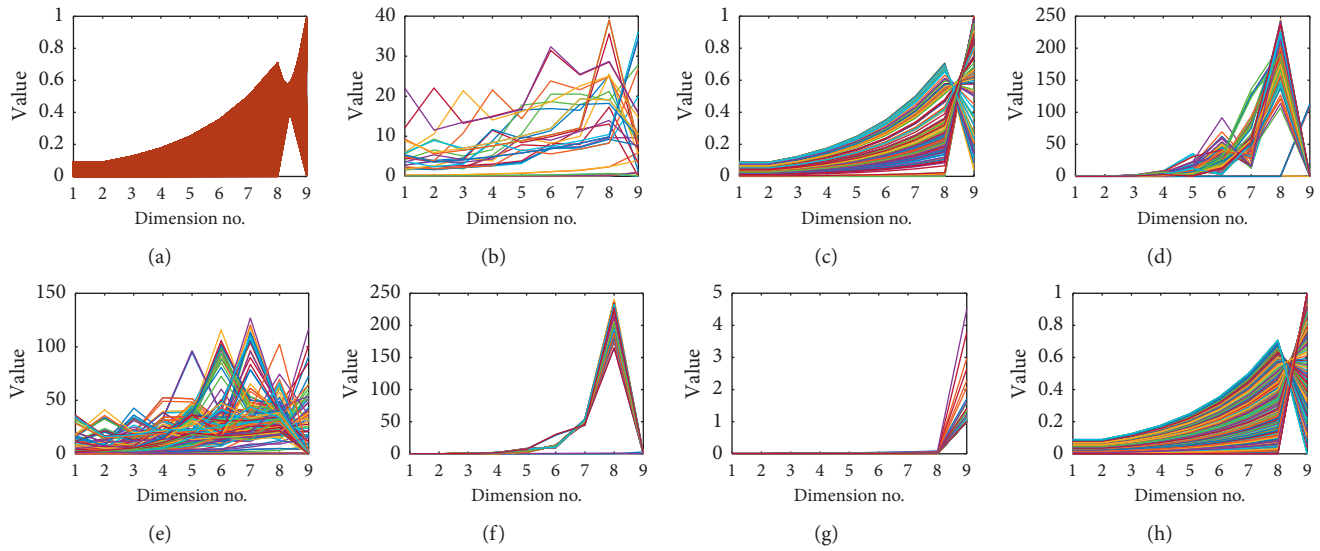


Figure 3: For 9-objective MaF6, its true PF and output populations of seven algorithms in parallel coordinate: (a) true PF of MaF6; (b) RVEA on MaF6; (c) NSGA-III on MaF6; (d) VaEA on MaF6; (e) MOEA/DD on MaF6; (f) SPEA2 + SDE on MaF6; (g) MOEA/D-AM2M on MaF6; (h) ASES on MaF6.

of the comparison algorithms RVEA, VaEA, MOEA/DD, SPEA2 + SDE, and MOEA/D-AM2M do not converge to the PF. Take the eighth optimization objective for instance, and the values of most output solutions from the above five algorithms are far greater than 0.7. The algorithm NSGA-III has better convergence and diversity than other five comparison algorithms. Comparing Figures 3(h) and 3(c), we can observe that the convergence of NSGA-III and ASES is similar, but the diversity of ASES is much better. The above comparison results demonstrate the competitive capability of the proposal in solving MaOPs having degenerate PFs.

The PF of test function MaF10 has complicated mixed geometries with scaled convex and concave segments. As shown in Figure 4, although the output population of algorithm ASES cannot fully approximate the true PF, combined with the distributions of the output populations and the HV/IGD values in Tables 1 and 2, we can derive that the proposed ASES is more powerful than the six comparison algorithms.

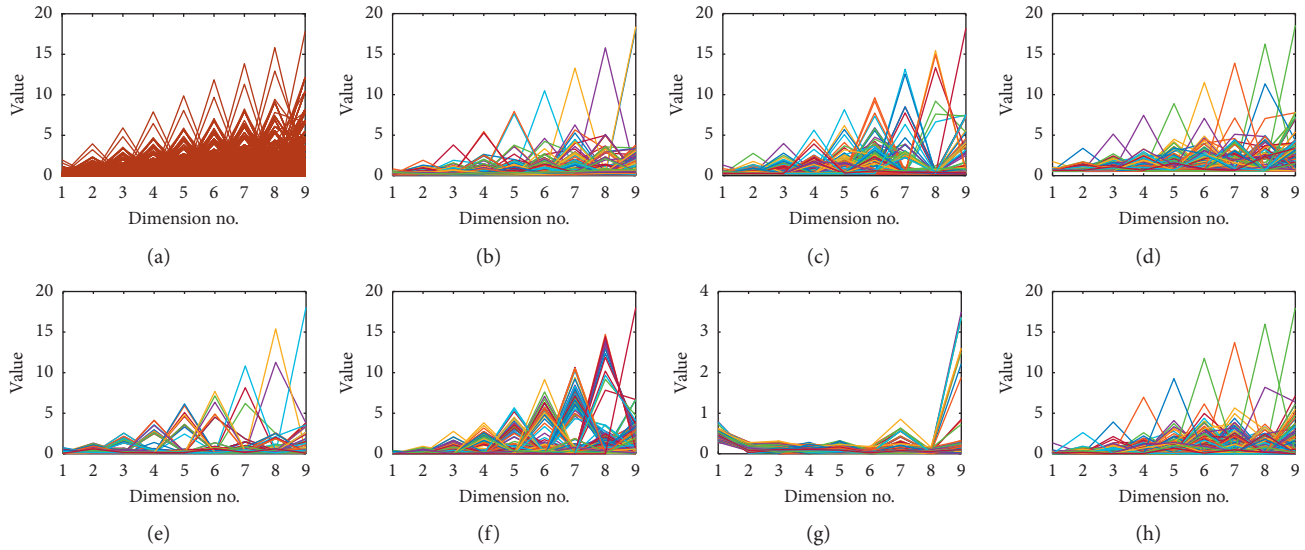


Figure 4: For 9-objective MaF10, its true PF and output populations of seven algorithms in parallel coordinate: (a) true PF of MaF10; (b) RVEA on MaF10; (c) NSGA-III on MaF10; (d) VaEA on MaF10; (e) MOEA/DD on MaF10; (f) SPEA2 + SDE on MaF10; (g) MOEA/D-AM2M on MaF10; (h) ASES on MaF10.

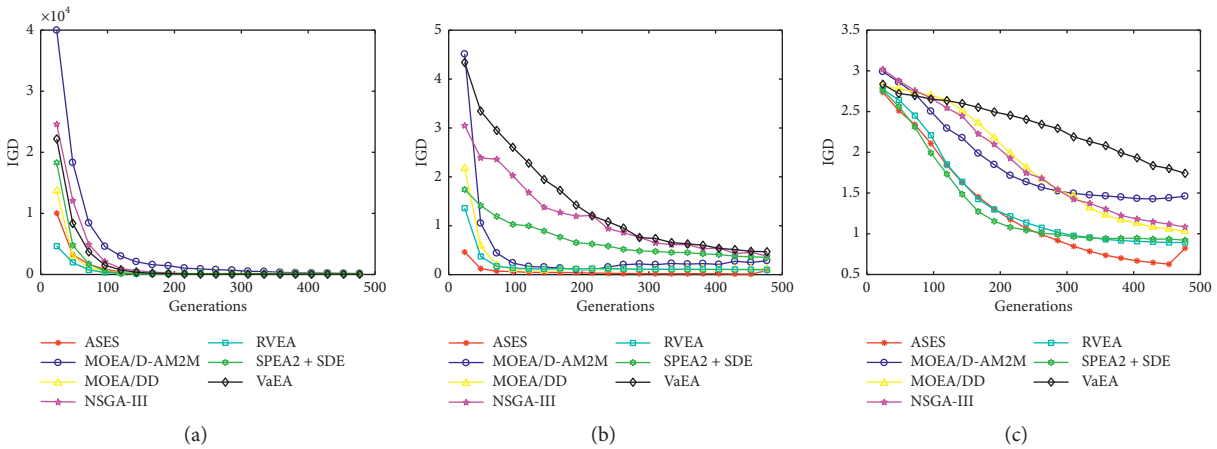


Figure 5: Convergence trend of six algorithms on 9-objective MaF4, MaF6, and MaF10.

Besides, the convergence trends of the six MaOEAs on solving MaF4, MaF6, and MaF10 with nine objectives with respect to metric IGD are illustrated in Figure 5. From Figure 5(a), we can observe that, in the initial stage, the IGD values of the proposed ASES decline slower than the RVEA. As the progress of optimization search, the proposed ASES still obtains lower IGD value lower than all the five comparison algorithms. For the 9-objective MaF6, as shown in Figure 5(b), the IGD of the proposed ASES declines much faster than all the five comparison algorithms. As shown in Figure 5(c), in the early stage, the convergence pace of the comparison algorithm SPEA2 + SDE is much faster than that of the algorithm ASES, but its IGD values remain unchanged after 300 generations. Then, the proposed ASES obtains lower IGD values than SPEA2 + SDE. The competitive performance of the ASES can be attributed to the fact that our proposed flexible framework has the capability to integrate the advantages of multiple MaOEAs.

The above experimental results have demonstrated that the proposed ensemble framework is highly beneficial to existing MaOEAs, and its prototype has competitive performance. To further investigate the behavior of the proposed ASES, the selection probability fluctuations of three variation operators during the entire evolutionary process for solving the 9-objective MaF1-MaF4 are plotted, as shown in Figure 6.

Figure 6 illustrates that no single mating and variation operator can dominate over the entire evolutionary process of solving any one instance. As the search progresses, the selection probabilities of different mating and variation operators fluctuate significantly. For instance, when solving 9-objective MaF1, at the beginning, the selection probability of operator from VaEA fluctuates between 0.35 and 0.43, while by about 400

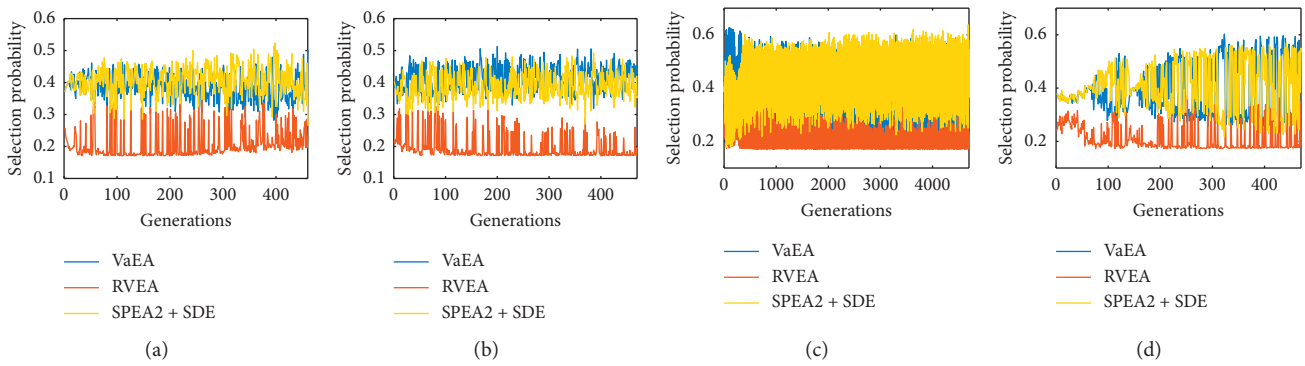


Figure 6: Selection probability fluctuations of three variation operators from the corresponding algorithms during evolution.

generations, its fluctuation range became 0.28 and 0.4. When we look at Figure 6(c), we find that the fluctuation characteristics of the three operators on 9-objective MaF3 are completely different from those the other three test instances. This is because test function MaF3 has a large number of local optimal solutions. These observations imply that the proposal can employ efficient operators according to the problem characteristics and different search stages.

5. Conclusions and Future Work

This paper focuses on the issue that no one single MOEA is suitable for solving various types of MaOPs. We design a flexible ensemble framework to integrate any number of MOEAs to promote their advantages to solve a wide range of MOPs. Besides, a big archive is embedded in this ensemble framework to alleviate the undesirable phenomenon that some promising solutions are discarded during the evolution process. Finally, extensive experiments are performed on 52 challenging test instances to verify the effectiveness of the proposal by comparing it with six state-of-the-art MaOEAs.

The MaOPs with complicated PSs are challenging the existing evolutionary multiobjective optimization algorithms. Solving these challenging problems is one of our next research. Also, applying evolutionary multiobjective optimization to solve problems coming from real-world applications, such as Internet of Things [65], resource investment project scheduling [66], supply chain management [67], and collaborative robots [68], is another interesting direction, especially designing problem-specific evolution rules and environment selection mechanisms. Besides, large-scale multiobjective optimization [69–72] is another challenging task in multiobjective optimization community and frequently occurs in real-world applications. Thus, designing efficient evolutionary algorithms for large-scale multiobjective optimization deserves future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 71871185.

References

[1] J. Xiong, C. Zhang, G. Kou, R. Wang, H. Ishibuchi, and F. E. Alsaadi, "Optimizing long-term bank financial products portfolio problems with a multiobjective evolutionary approach," *Complexity*, vol. 2020, pp. 1–18, 2020.

[2] J. Xiong, R. Wang, and J. Jiang, "Weapon selection and planning problems using moea/d with distance-based divided neighborhoods," *Complexity*, vol. 2019, Article ID 7589760, 18 pages, 2019.

[3] J. Xiong, R. Leus, Z. Yang, and H. A. Abbass, "Evolutionary multi-objective resource allocation and scheduling in the Chinese navigation satellite system project," *European Journal of Operational Research*, vol. 251, no. 2, pp. 662–675, 2016.

[4] R. Wang, J. Xiong, M.-f. He, L. Gao, and L. Wang, "Multi-objective optimal design of hybrid renewable energy system under multiple scenarios," *Renewable Energy*, vol. 151, pp. 226–237, 2020.

[5] Q. Zhu, "Research on road traffic situation awareness system based on image big data," *IEEE Intelligent Systems*, vol. 35, no. 1, pp. 18–26, 2019.

[6] Y. Liu, C. Yang, and Q. Sun, "Thresholds based image extraction schemes in big data environment in intelligent traffic management," *IEEE Transactions on Intelligent Transportation Systems*, 2020, In press.

[7] B. Cao, J. Zhao, P. Yang et al., "Multiobjective 3-D topology optimization of next-generation wireless data center network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3597–3605, 2019.

[8] B. Cao, J. Zhao, Y. Gu, S. Fan, and P. Yang, "Security-aware industrial wireless sensor network deployment optimization," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5309–5316, 2019.

[9] J. Yan, W. Pu, S. Zhou, H. Liu, and Z. Bao, "Collaborative detection and power allocation framework for target tracking in multiple radar system," *Information Fusion*, vol. 55, pp. 173–183, 2020.

[10] J. Yan, W. Pu, S. Zhou, H. Liu, and M. S. Greco, "Optimal resource allocation for asynchronous multiple targets tracking in heterogeneous radar networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4055–4068, 2020.

[11] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Transactions on Services Computing*, p. 1, 2019, In press.

[12] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proceedings Of Annual Meeting Of the North American on Fuzzy Information Processing Society*, pp. 233–238, IEEE, New Orleans, LA, USA, 2002.

[13] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2012.

[14] F. Wang, Y. Li, F. Liao, and H. Yan, "An ensemble learning based prediction strategy for dynamic multi-objective optimization," *Applied Soft Computing*, vol. 96, Article ID 106592, 2020.

[15] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[16] B. Cao, X. Wang, W. Zhang, H. Song, and Z. Lv, "A many-objective optimization model of industrial internet of things based on private blockchain," *IEEE Network*, vol. 34, no. 5, pp. 78–83, 2020.

[17] B. Cao, W. Dong, Z. Lv, Y. Gu, S. Singh, and P. Kumar, "Hybrid microgrid many-objective sizing optimization with fuzzy decision," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 11, pp. 2702–2710, 2020.

[18] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[19] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[20] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: a comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26 194–26214, 2018.

[21] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2016.

[22] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[23] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[24] F. Li, R. Cheng, J. Liu, and Y. Jin, "A two-stage R2 indicator based evolutionary algorithm for many-objective optimization," *Applied Soft Computing*, vol. 67, pp. 245–260, 2018.

[25] J. G. Falcón-Cardona and C. A. C. Coello, "Indicator-based multi-objective evolutionary algorithms: a comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, 2020.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[27] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, 2013.

[28] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2013.

[29] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.

[30] C. Wang, R. Xu, J. Qiu, and X. Zhang, "AdaBoost-inspired multi-operator ensemble strategy for multi-objective evolutionary algorithms," *Neurocomputing*, vol. 384, pp. 243–255, 2020.

[31] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: an improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2014.

[32] Y.-H. Zhang, Y.-J. Gong, T.-L. Gu, and J. Zhang, "Ensemble mating selection in evolutionary many-objective search," *Applied Soft Computing*, vol. 76, pp. 294–312, 2019.

[33] H. Ishibuchi, L. M. Pang, and K. Shang, "A new framework of evolutionary multi-objective algorithms with an unbounded external archive," Technical report, 2020.

[34] Y. Nan, K. Shang, H. Ishibuchi, and L. He, "Reverse strategy for non-dominated archiving," *IEEE Access*, vol. 8, pp. 119 458–119 469, 2020.

[35] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2016.

[36] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[37] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[38] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 400–413, 2014.

[39] M. Pal, S. Saha, and S. Bandyopadhyay, "DECOR: differential evolution using clustering based objective reduction for many-objective optimization," *Information Sciences*, vol. 423, pp. 200–218, 2018.

[40] J. Zhou, X. Yao, Y. Lin, F. T. S. Chan, and Y. Li, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Information Sciences*, vol. 456, pp. 50–82, 2018.

[41] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.

[42] F. Wang, Y. Li, A. Zhou, and K. Tang, "An estimation of distribution algorithm for mixed-variable newsvendor problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 479–493, 2020.

[43] Q. Lin, S. Liu, Q. Zhu et al., "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32–46, 2016.

[44] S. M. Venske, R. A. Gonçalves, and M. R. Delgado, "ADEMO/D: multiobjective optimization by an adaptive differential evolution algorithm," *Neurocomputing*, vol. 127, pp. 65–77, 2014.

[45] W. Wang, S. Yang, Q. Lin et al., "An effective ensemble framework for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 645–659, 2019.

[46] A. Santiago, B. Dorronsoro, A. J. Nebro, J. J. Durillo, O. Castillo, and H. J. Fraire, "A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: fame," *Information Sciences*, vol. 471, pp. 233–251, 2019.

[47] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2013.

[48] Q. Xu, Z. Xu, and T. Ma, "A survey of multiobjective evolutionary algorithms based on decomposition: variants, challenges and future directions," *IEEE Access*, vol. 8, pp. 41588–41614, 2020.

[49] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, "Hyperplane assisted evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3367–3380, 2020.

[50] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 247–263, 2017.

[51] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2014.

[52] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[53] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, In press.

[54] Z. Liang, T. Luo, K. Hu, X. Ma, and Z. Zhu, "An indicator-based many-objective evolutionary algorithm with boundary protection," *IEEE Transactions on Cybernetics*, 2020, In press.

[55] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 6, pp. 2193–2196, 2012.

[56] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking multi- and many-objective evolutionary algorithms under two optimization scenarios," *IEEE Access*, vol. 5, pp. 19 597–19 619, 2017.

[57] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172–186, 2018.

[58] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[59] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[60] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[61] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[62] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[63] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[64] M. Li, L. Zhen, and X. Yao, "How to read many-objective solution sets in parallel coordinates [educational Forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 88–100, 2017.

[65] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on sdn and nfv for next generation iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706–5712, 2020.

[66] J. Xiong, J. Liu, Y. Chen, and H. A. Abbass, "A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 742–763, 2014.

[67] F. Wang, A. Diabat, and L. Wu, "Supply chain coordination with competing suppliers under price-sensitive stochastic demand," *International Journal of Production Economics*, Article ID 108020, 2021, In press.

[68] Z. Lv and L. Qiao, "Deep belief network and linear perceptron based cognitive computing for collaborative robots," *Applied Soft Computing*, vol. 92, p. 106300, 2020.

[69] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Information Sciences*, vol. 509, pp. 457–469, 2020.

[70] B. Cao, S. Fan, J. Zhao, P. Yang, K. Muhammad, and M. Tanveer, "Quantum-enhanced multiobjective large-scale optimization via parallelism," *Swarm and Evolutionary Computation*, vol. 57, Article ID 100697, 2020.

[71] B. Cao, J. Zhao, Y. Gu, Y. Ling, and X. Ma, "Applying graph-based differential grouping for multiobjective large-scale optimization," *Swarm and Evolutionary Computation*, vol. 53, Article ID 100626, 2020.

[72] C. He, L. Li, Y. Tian et al., "Accelerating large-scale multi-objective optimization via problem reformulation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 949–961, 2019.

Hindawi
Complexity
Volume 2021, Article ID 6676934, 15 pages
https://doi.org/10.1155/2021/6676934

Research Article

A Chaotic Disturbance Wolf Pack Algorithm for Solving Ultrahigh-Dimensional Complex Functions

Qiming Zhu, Husheng Wu, Na Li, and Jinqiang Hu

College of Equipment Support and Management, Engineering University of PAP, Xi'an 710086, China

Correspondence should be addressed to Husheng Wu; wuhusheng0421@163.com

Received 7 November 2020; Revised 11 January 2021; Accepted 2 February 2021; Published 12 February 2021

Academic Editor: Rui Wang

The optimization of high-dimensional functions is an important problem in both science and engineering. Wolf pack algorithm is a technique often used for computing the global optimum of a multivariable function. In this paper, we develop a new wolf pack algorithm that can accurately compute the optimal value of a high-dimensional function. First, chaotic opposite initialization is designed to improve the quality of initial solution. Second, the disturbance factor is added in the scouting process to enhance the searching ability of wolves, and an adaptive step length is designed to enhance the global searching ability to prevent wolves from falling into the local optimum effectively. A set of standard test functions are selected to test the performance of the proposed algorithm, and the test results are compared with other algorithms. The high-dimensional and ultrahigh-dimensional functions (500 and 1000) are tested. The experimental results show that the proposed algorithm features in good global convergence, high accuracy calculation, strong robustness, and excellent performance in high-dimensional functions.

1. Introduction

The rapid development of science and technology enriches human life. Meanwhile, the problems encountered in various fields, especially in engineering and technology related fields, become increasingly complex and diversified, for example, large-scale dispatching problems [1–3] and large-scale power system optimization problems [4–6]. The complexity of the problem increases exponentially with the increase of dimension, which reduces the performance of traditional optimization algorithms and thus the ability to solve this kind of problems well. Inspired by the living mode of creatures in nature, swarm intelligence algorithm contains the wisdom of nature for thousands of years. Compared with the traditional optimization algorithm, its structure is simple and easy to implement, so it is widely used. Swarm intelligence algorithm is applied to high-dimensional function optimization problems by many scholars. The study in [7] used neighborhood factors to improve the NFO algorithm, which is applied to complex optimization problems and achieves good results. Compared with the traditional particle swarm optimization (PSO) algorithm, the improved particle swarm optimization algorithm in [8] is applied to high-dimensional function optimization because its convergence accuracy in high-dimensional function is improved. The study in [9] applied the improved dolphin colony algorithm to solve the high-dimensional function optimization problem and achieved good results. Although the performance of the improved method outweighs the original algorithm in high-dimensional function optimization, the improved one still has low convergence precision and low solving efficiency, making it difficult to meet the requirements of more and more refinement in the field of engineering science and technology.

In [10], wolf pack algorithm (WPA) was proposed based on a detailed analysis of wolves' scouting behavior and prey allocation methods. WPA is abstracted as three kinds of artificial wolves (the lead wolf, the scout wolf, and the ferocious wolf). In the process of hunting, there are three kinds of intelligent behaviors (the calling of the lead wolf, the wandering of the scout wolf, and the siege of the ferocious wolf), in addition to the generation rule of the "winner is the king" and the population updating rule of "survival of the fittest". As a novel swarm intelligence algorithm, WPA has good performance in global optimization and local exploration [11]. Its optimization method is different from the

previous bionic intelligent algorithms such as particle swarm optimization (PSO) [12], ant colony optimization (ACO) algorithm [13], shuffled frog leaping algorithm (SFLA) [14], genetic algorithm (GA) [15], and artificial bee colony (ABC) algorithm [16]. It is widely adopted in PID parameter tuning [17], path planning [18], knapsack problem [19], combinational optimization problem [20], etc. The study in [10] indicates that the superiority of WPA is obvious in solving complex and high-dimensional problems. However, some shortcomings still exist in the algorithm: for example, the initial population produced randomly makes the algorithm easily fall into local optimum, and low quality initial solution increases the calculation amount of the algorithm; the greedy search method and rigid wandering direction of wolf detection stand great chances of falling into local optimum and missing the optimal solution; the step size of three linear relationships leads to the inflexible movement of the whole wolf pack, increasing the computational complexity of the algorithm.

The study in [21] makes use of tent chaotic mapping method to improve the quality of initial solution to the wolf pack algorithm, which endows the WPA with faster convergence speed and higher solution accuracy. In [22], a chaos optimization method based on logistic map was used to initialize the population, which improved the optimization accuracy and convergence rate of WPA. In [23], the d-dimensional chaotic variables were mapped to the solution space to obtain the initial wolf group. With the aid of opposition based learning (OBL), the study in [24] adopts an opposite wolf pack initialization method and puts forward an oppositional wolf pack algorithm (OWPA), which improves the quality of the initial solution and the convergence speed of the algorithm.

However, considering only the initial population quality or initial population distribution, the above method is still prone to fall into local optimum and the convergence accuracy is not high in some functions. Meanwhile, the above improved method is not applied to high-dimensional function optimization, so it is difficult to show that it has advantages in high-dimensional function optimization.

Compared with the existing research work, the four main innovative aspects of this paper are as follows:

(1) Chaotic self-logical sequences are used and mapped to the solution space, and then the initial population is further optimized by the reverse formula

(2) Disturbance factor is used to disturb the wolf 's scout direction to increase the randomness of the scout

(3) The scouting step length of the scout wolf and the ferocious wolf is set as the adaptive step size, so as to adjust the moving speed according to their own position during the movement process

(4) A lot of experimental comparisons are made on hyperdimensional function showing the effectiveness of the improved algorithm

The remainder of this paper is organized as follows: In Section 2, the chaotic disturbance wolf pack algorithm (CDWPA) is described in detail. In Section 3, the theoretical analysis of the algorithm is provided. In Section 4, a variety of benchmark functions with different mathematical characteristics are tested. The simulation turns out that the CDWPA possesses higher accuracy in convergence and computational robustness and can effectively solve the problem of high-dimensional function optimization.

2. Chaotic Disturbance Wolf Pack Algorithm

The wolves are located in the Euclidean space $N \times D$, in which N is the total number of artificial wolves in the pack and $D$is the number of variables to be optimized. The position of the artificial wolf X_i can be expressed as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, where X_{id} is the position of the artificial wolf X_i in the d-dimensional variable space$d = (1, 2, \ldots, D)$ to be optimized; the odor concentration of prey perceived by the artificial wolf X_i is $Y_i = f(X_i)$, where Y_i is the value of the target function. The improved wolf pack algorithm is composed of chaos reverse initialization population, wolf generation, adaptive and disturbance scout, fierce wolf siege, and population update rules. The algorithm flowchart is shown in Figure 1.

2.1. Chaotic Opposite Initialization of the Population. Chaotic state exists widely in nature and society. It has the characteristics of randomness, ergodicity, and regularity. Chaotic movement can traverse all the states according to its own law in a certain range. Therefore, in order to ensure the diversity and randomness of the individual population of wolves X_i, this paper adopts logical self-mapping (LSM) function to generate the wolf pack sequence. First, random wolves X_i were created, and then wolves X_i were mapped to chaotic space $(-1, 1)$ by

$$Z_i = 2 \times \frac{X_i - \text{lb}}{\text{ub} - \text{lb}} - 1, \tag{1}$$

$$Z_{i+1} = 1 - 2 \times Z_i^2. \tag{2}$$

To intuitively display the characteristics of the above chaotic maps, the probability distribution characteristics of the logical self-mapping on the (0, 1) interval are compared with those of Gauss map [25], sinusoidal iterator [26], and logistic map [27]. The calculation of probability distribution is estimated as follows: 50000 chaotic numerical points are generated by 50000 iterations of chaotic map, the interval (0, 1) is equally divided into 100 intervals, and the probability of these 50000 points falling into 100 equal intervals is counted; the probability distribution diagram is shown in Figures 2–5.

From the analysis of Figures 2–5, the distribution of sinusoidal iterator is low in the front end but high in the back end, and most of the initial solutions are distributed in the interval $(0.5, 1)$. If the optimal value of the function problem does not exist in this interval, it will easily miss the optimal solution by the sinusoid iterator. The Gauss map is just the opposite; its probability distribution shows a decreasing trend, and the distribution trend of the logistic map is very large in middle and small at both ends. If the global optimal solution of the function problem is distributed in the middle,

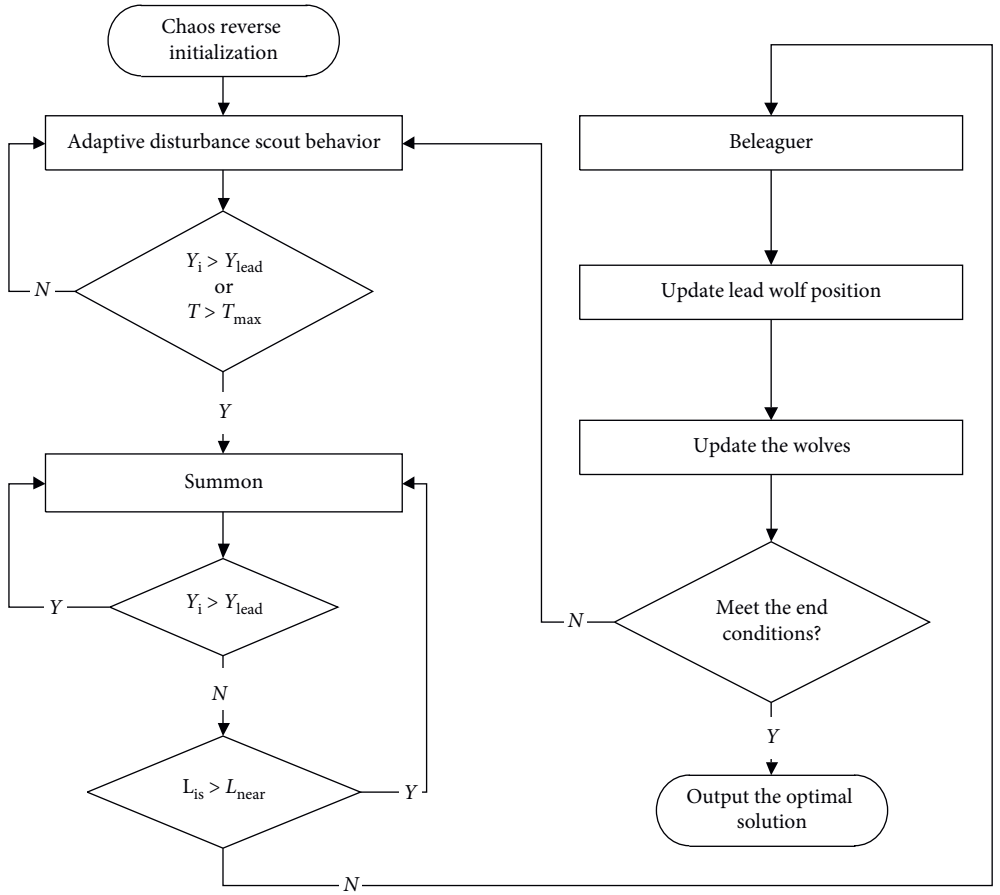


Figure 1: Flowchart of CDWPA.

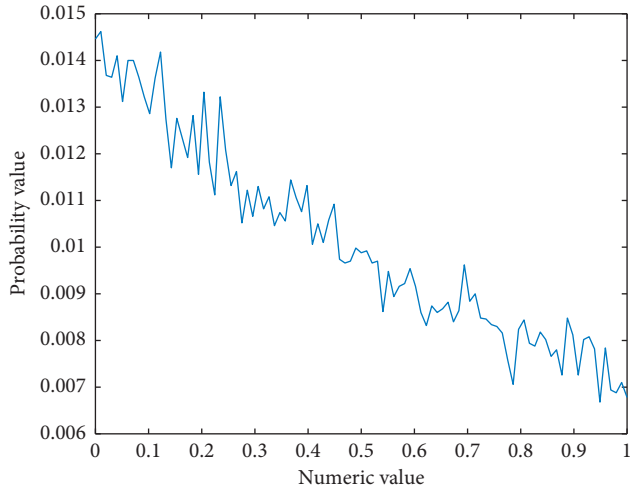


Figure 2: Gauss map.

a large number of invalid searches will appear, which will be unfavorable to global optimization. LSM chaotic sequences are uniformly distributed in the interval (0, 1). This distribution can avoid oversearching in some local areas (for example, a large number of searches of logistic maps are concentrated at both ends of the interval), thus reducing the adverse effects on the optimization algorithm due to the mismatch between the distribution characteristics of chaotic sequences and the position of the global optimal solution of the optimization problem.

If the definition domain of logical self-mapping function is $(-1, 1)$ except for 0 and 0.5, chaos will take effect in logic definition. After the chaotic variable sequence is obtained by chaotic search, the chaotic ergodic sequence should be

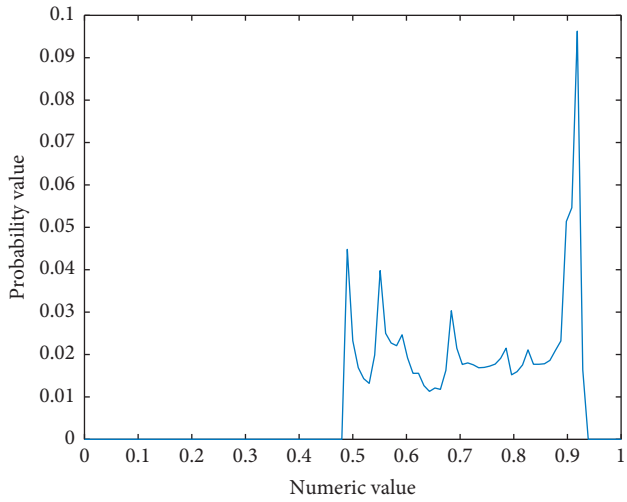


Figure 3: Sinusoidal iterator.

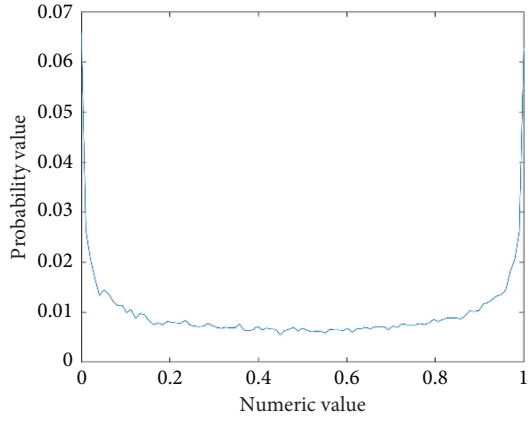


Figure 4: Logistic map.

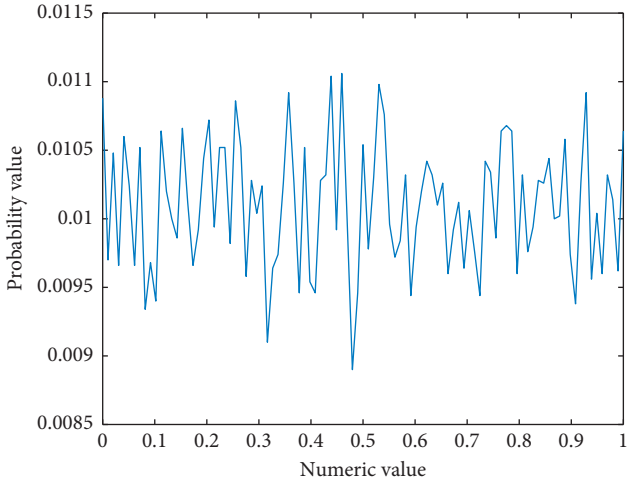


Figure 5: LSM.

transformed into the original solution space according to the following formula to evaluate the fitness value.

$$X_{i+1} = 0.5 \times (\text{ub} - \text{lb}) * Z_{i+1} + 0.5 \times (\text{ub} + \text{lb}). \quad (3)$$

Among them, uband lbrepresent the upper and lower boundaries of the artificial wolf X_i, respectively. Besides, the objective function values of the system random wolf pack X_i and chaotic mapping wolf pack X_{i+1} are calculated severally. Once the better solution is found in this process, the better wolf pack will replace the original wolf pack. Then the optimized wolf pack is operated reversely to calculate the reverse wolf pack.

$$OX_i' = L_d + U_d - x_{i+1}. \quad (4)$$

Many artificial wolves with better target function values are selected from the $\{X_{i+1} \cup OX'\}$ to form the initial wolves and complete the initialization of the wolves.

2.2. Selecting Lead Wolf. The lead wolf, responsible for directing the whole to cooperate in hunting, is the artificial wolf with the largest odor concentration of prey in each generation. In the algorithm, the artificial wolf with the largest objective function value is selected as the lead wolf in the current wolf pack. There is only one lead wolf. If there are multiple wolves with the same objective function value, one of them will be randomly selected as the leader. Meanwhile, the lead wolf is not a life-long system. In the process of operation, if the objective function value of the artificial wolf X_i is better than that of the current one, the artificial wolf X_i will replace the former one and become the current leader. The lead wolf does not perform the behavior of wandering, besieging, and so on.

2.3. Adaptive Disturbance Scout. After the lead wolf is determined, the scout wolf proceeds with optimization according to its own position, and the basic calculation formula is the following one:

$$x_{id} = x_{id} + \text{step} \times \sin\left(2\pi \times \frac{p}{h}\right). \quad (5)$$

In this formula, x_{id} refers to the place of the current searching wolf, stepis the walk length, h is the walk direction, $p \in (1, h)$. After h is given in the formula, the searching wolf can only walk in a fixed direction, which is easy to fall into local optimum. Therefore, a random disturbance factor $\delta = \text{rand}(\text{round}(p/p))$ is designed to change the walking formula to

$$x_{id} = x_{id} + \text{step} * \sin\left(2\pi \times \frac{p}{h} + \delta\right). \quad (6)$$

At the same time, considering the distance between the detection wolf and the lead wolf, the scouting step length is improved to

$$\text{step} = \text{rand} \times \text{norm}\left\|x_i - x_{best}\right\|, \quad (7)$$

where x_i stands for the current location of the lead wolf and x_{best} is the optimal location of the lead wolf, so that the detection wolf can not only cover the fixed search range previously, but also search other directions randomly, which increases the optimal searching ability by breaking through the encirclement. The walking location set of the detection

wolf is $x_{id} = \{x_{id}^1, x_{id}^2, \ldots, x_{id}^h\}$, and the value of the target function and that of the lead wolf are analyzed. If the fitness value is greater than the current lead wolf, then the current wolf detector becomes the lead wolf.

By analyzing (5), after the search direction h is fixed, the migration of different wolf detectors is equivalent to the movement in fixed directions of h. As shown in Figure 6, if the optimal prey is located at the parallel lines of the search direction, the detection wolf will be easily trapped into the local optimum and escape will be difficult. When the disturbance factor δ and adaptive step size are added, the search direction of the wolf detector is more random and possesses strong ability to break through; thus, it is not easy to fall into local optimum.

2.4. The Besiege of Ferocious Wolf. When the scouting behavior is over, the current artificial lead wolf sends out a calling behavior to inform the other artificial wolves of their own location and fitness value. The rest of the artificial wolves move closer to the lead wolf in a rush step according to their distance from the lead wolf.

$$x_{id}^{k+1} = x_{id}^k + \text{step}_b^d \times \frac{g_d^k - x_{id}^k}{\left|g_d^k - x_{id}^k\right|}. \tag{8}$$

In the formula, g_d^k stands for the current location of the lead wolf. step_b^d is the besieging size of ferocious wolf. When the artificial wolf is far away from the lead wolf, it approaches the lead wolf with a larger step length; when the artificial wolf is close to the lead wolf, it approaches the lead wolf with a smaller step length. In addition, if the odor concentration of prey perceived by the *ferocious* wolf is greater than that of the current lead wolf, the strong one will become the leader, guiding the wolf pack to hunt until it is replaced by a better wolf or its operations is over.

After the iteration of the $k + 1$, if the odor concentration of prey perceived by wolf W_i is $Y_i^{k+1} > Y_{\text{lead}}^k$, then the ferocious wolf W_i replaces the former wolf leader and initiates the calling behavior. If the odor concentration of prey perceived by wolf W_i is $Y_i^{k+1} < Y_{\text{lead}}^k$, then the ferocious wolf W_i will continue to attack until the distance L_{is} between wolf W_i and the lead wolf is less than the determined distance L_{near}; it changes into besiege behavior. Given that the value range of the d-variable to be optimized is $[\min_d, \max_d]$, ω is the determinant factor in distance, and the determinant distance is

$$L_{\text{near}} = \frac{1}{D\omega} \sum_{d=1}^{D} \left|\max_d - \min_d\right|. \tag{9}$$

The Act of Siege. After the attack, the wolves besiege the prey to capture it. At this time, the position of the wolf is regarded as the location of the prey. If the prey's position in the d-dimensional space is G_d^k, the siege behavior of wolves can be expressed by the following equation:

$$x_{id}^{k+1} = x_{id}^k + \lambda \cdot \text{step}_c^d \cdot \left|G_d^k - x_{id}^k\right|, \tag{10}$$

where λ is the random integer evenly distributed in the interval $[-1, 1]$ and step_c is the siege step length. If the odor concentration of prey perceived by artificial wolf is greater than that of its original position after the siege, the location of the artificial wolf will be updated; otherwise, the location of the artificial wolf will remain unchanged.

2.5. Updating the Wolves. To maintain the competitiveness and diversity of the wolf pack, after the end of the hunting behavior, the wolf pack will distribute the prey according to their contributions and eliminate the artificial wolf R with the worst fitness value. At the same time, the artificial wolf R will be randomly generated to complete the population updating. R is the random integer in $[(N/2\beta), (N/\beta)]$, and β will be taken as the updating proportion factor.

The steps of chaotic disturbance wolf pack algorithm are as follows:

Step 1. Chaos reverse wolf group initialization: initialize the spatial coordinates of wolf pack randomly in the solution space according to (1)–(4).

Step 2. Adaptive disturbance walk: the walk strategy is carried out on (6)–(7).

Step 3. Calling behavior: the lead wolf sends out signals based on its position; the besieging wolf takes adaptive attack strategy to approach the lead wolf after receiving the signal.

Step 4. The siege of ferocious wolf: after reaching the siege distance, the ferocious wolf starts to encircle the prey.

Step 5. Update the location of the wolf pack: sort on the basis of the fitness value of the wolf pack, and select the best one.

Step 6. Population regeneration: individuals with poor fitness in the population will be eliminated, and an equal number of new individuals will be generated.

Step 7. Repeat the process of 2–6 until the maximum number of iterations is reached or the algorithm accuracy meets the threshold.

3. Theoretical Analysis of the CDWPA Algorithm

3.1. Convergence Analysis of the Algorithm. Markov chain is a stochastic process with no aftereffect, which is often used to prove the convergence of the algorithm [28]. CDWPA is a process that repeatedly produces chaotic sequence, reverse scouting, summoning, rush, besieging, and wolf pack renewal behavior. Each behavior of the population is only related to the current state of the population but has nothing to do with the previous state. Therefore, the CDWPA population sequence conforms to Markov chain.

Reference [29] has proved that if an intelligent algorithm satisfies the following two conditions, the intelligent algorithm converges to the global optimal solution with probability 1.

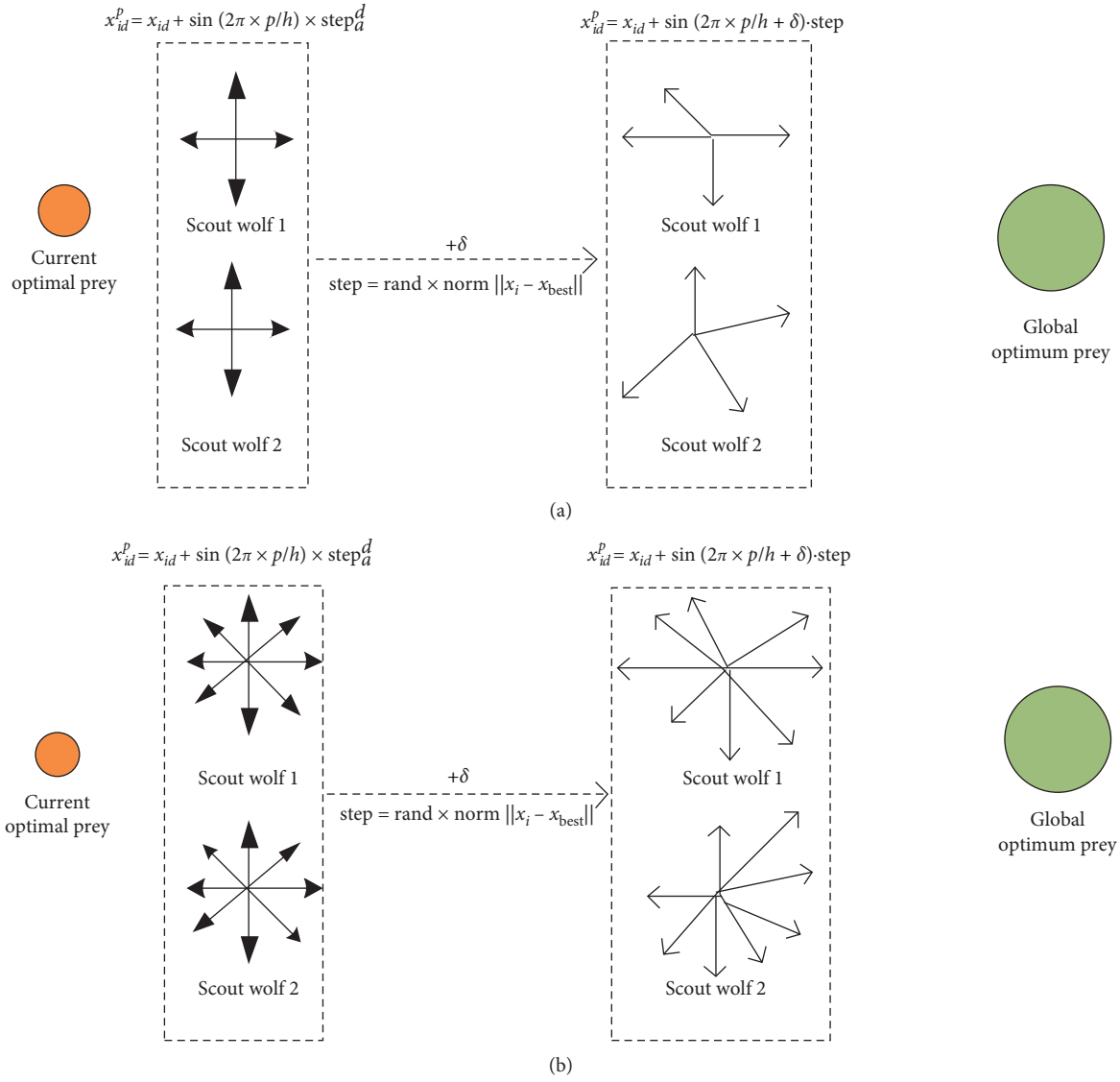


Figure 6: Disturbed scout and fixed scout: (a) when $h = 4$, $p = 1:4$; (b) when $h = 8$, $p = 1:8$.

Condition one: the sum of any two solutions x_1 and x_2 in the solution space x_2 can be obtained by various operators in x_1 of the algorithm

Condition two: if the population sequence $Q_1, Q_2, \ldots, Q_N$ is monotone

According to relevant theories, the problem of proof is transformed into the following two points:

Point one: the CDWPA population sequence is ergodic

Point two: CDWPA is a finite homogeneous Markov chain

If the above two conditions are satisfied, CDWPA will converge to the global optimal solution with probability 1.

Assuming that the search space of CDWPA is Q and the state changes of state space caused by chaotic sequence, reverse population, scouting, rushing, besieging, and population updating are, respectively, expressed by transition probability H, F, Y, B, W, Z, then the transition probability matrix of Markov chain of CDWPA is obtained as

$$P = H \times F \times Y \times B \times W \times Z. \tag{11}$$

The definitions are given as follows:

Definition 1. Assume the P_{ij} to be the transition probability matrix of a Markov chain. If there is such a matrix for $i, j \in Q \ni k \geq 1$, resulting in $P_{ij}^{k} > 0$, then the Markov chain is said to be irreducible.

Definition 2. Assume that there is a nonempty set $I = \{k \mid k \geq 1, P_{ij}^k > 0, \forall i, j \in Q\}$. If the greatest common divisor of the set I is 1, then the Markov chain is aperiodic.

Definition 3. Assume that $u_i = \sum_k^{\infty} k \cdot P_{ij}^k$ for a recurrent state i. If $u_i < +\infty$, u_i is called normal return. In particular, if i is normal recurrent and aperiodic, then the Markov chain is ergodic.

To prove the convergence of CDWPA, we have the following two points.

Point one: the Markov chain of CDWPA population sequence is ergodic.

The following is proved: (1) Markov chain of CDWPA population sequence is irreducible. Suppose the generation population k of the algorithm is $Q_k = \{X_1, X_2, \ldots, X_N\}$, X_i is the state of the first artificial wolf. Because the transition probability matrix $P_{ij} = P\{Q_{k+1} = j \mid Q_k = i, k \geq 1\}$ of the population Markov chain is only related to the initial and terminal states i, j and $Q_k > 0$ is always true, the transition probability matrix P of a population is positive definite matrix. According to Definition 1, the Markov chain of CDWPA population sequence is irreducible. (2) The Markov chain of CDWPA population is aperiodic and irreducible. For a given $k > 0$, the Markov chain obtained from condition one is irreducible, and $\ni j \in Q$ is inevitable, so that it holds that $P_{ij} > 0$, and combined with Definition 2, $k = 1$. Therefore, the maximum common divisor of Q is 1, so the Markov chain of CDWPA population is aperiodic. From (1) The Markov chain of CDWPA population is aperiodic and irreducible.

The Markov chain of CDWPA population is ergodic.

Since the values of the transition matrix H, F, Y, B, W, and Z are all within [0, 1] and P_{ij} is the probability of the state i being transformed into a state j through various behaviors, there must be $0 \leq P_{ij} \leq 1$; let $\sigma = \max\{P_{ij} \mid i, j \in \Omega\}$. The *Cauchy–Riemann equation* and Definition 3 prove $u_i = \sum_k^{\infty} k \cdot P_{ij}^k \leq \sum_k^{\infty} k\varepsilon^k \leq \infty$.

In conclusion, the Markov chain of CDWPA population sequence is ergodic, and point one is proved.

Point two: CDWPA is a finite homogeneous Markov chain.

It is proved that each generation of population Q_k is limited, and so is Markov chain; secondly, the algorithm repeatedly generates chaotic sequence, reverse scouting, summoning, rush, siege, and wolf pack update behavior to find better prey, and the individual renewal has the characteristics of high-quality selection. The generation of population is only related to current generation. After repeated iterative optimization, CDWPA can obtain a set of sequence solutions, and the sequence is a finite homogeneous Markov chain. To sum up, reasoning 1 and 2 show that the Markov chain of CDWPA population sequence is ergodic, and its sequence solution is a finite homogeneous Markov chain. Therefore, it is proved that CDWPA converges to the global optimal solution of the problem with probability 1.

3.2. Time Complexity Analysis of the Algorithm. The time complexity reflects the operation efficiency of the algorithm. The time complexity of cuckoo algorithm is analyzed in [30]. In this paper, the same method is adopted to analyze the time complexity of CDWPA.

In the WPA, we set the size of the wolf pack as N and the individual dimension as n. If the time of step size step, update scale factor β, search direction of wolf $T_{\max}$, and judging distance d_{near} is η_0, the time of generating random number is η_1, and the time of solving fitness function value is $f(n)$, then the time complexity of initialization phase is shown in

$$O(\eta_0 + N(n \cdot \eta_1 + f(n))) = O(n). \tag{12}$$

The wolf pack was sorted in line with the fitness value, the time of the lead wolf was selected as η_2, and the time for the artificial wolf to execute the walking strategy was calculated as η_3. In the summoning step, the distance between the artificial wolf X_i and the lead wolf X_{lead} is η_4, the time needed to move the position of a single wolf in each dimension is η_5, and the time to judge whether the attack range is reached is η_6; then, the time complexity of this stage is given as

$$O(N(\eta_2 + \eta_3 + \eta_4 + n \cdot \eta_5 + \eta_6)) = O(n + f(n)). \tag{13}$$

From the above formula, we can get the time complexity of the (WPA) to obtain the optimal solution of each generation.

$$T_n = O(n) + O(n + f(n)) = O(n + f(n)). \tag{14}$$

The process of CDWPA is analyzed. In the initialization stage, the generation time of logical self-mapping sequence is t_1, and the time of reverse generation population is t_2. The other generation parameters, dimensions, and solution fitness function values are the same as those of WPA. Then, the time complexity of CDWPA initialization phase is given as

$$O(\eta_0 + N(n \cdot \eta_1 + f(n) + t_1 + t_2)) = O(n + f(n)). \tag{15}$$

The time of calculating the distance between the detection wolf X_i and the lead wolf added by the adaptive step length is η_7, and the other wolves are selected to perform the same steps as WPA, such as scouting, summoning, and besieging; then, the time complexity of this stage is

$$O(N(\eta_2 + \eta_3 + \eta_4 + n \cdot \eta_5 + \eta_6 + \eta_7)) = O(n + f(n)). \tag{16}$$

To sum up, the total time complexity of CDWPA to find the optimal value of each generation is shown as

$$O(n) + O(n + f(n)) = O(n + f(n)). \tag{17}$$

From the above analysis, compared with WPA, the time complexity of CDWPA does not change, and the efficiency of CDWPA does not decrease.

4. Simulation Experiment and Algorithm Validity Test

4.1. Test Function. In order to test the performance of the CDWPA proposed in this paper, we select 10 standard test functions in [31] (dimensions from 2 to 200) to conduct the first set of experiments, and compare the optimization performance with wolf pack algorithm (WPA), oppositional wolf pack algorithm (OWPA) in [24], chaotic wolf pack algorithm (CWPA) in [22], PSO, ABC, and ASFA [32]. In order to further verify the ability of the improved algorithm to solve high-dimensional complex functions, the second set of experiments was carried out. Four high-dimensional (500 and 1000 dimensions) functions covering different types were selected, and the test results of seven algorithms were compared. The specific characteristics of the 10 standard test functions are shown in Table 1. "U" is unimodal function, "M" is multimodal function, "S" is separable function, and "N" is nonseparable function. Unimodal function is used to test the mining ability of the algorithm, multimodal function tests the exploration ability of the algorithm, and high-dimensional function tests the ability of the algorithm to solve complex problems.

To express the optimization effect more clearly, BEST, WORST, MEAN, and STD are selected to represent the best objective function value, the worst value, the mean value of target function, and the standard deviation of target function at the end of the optimization result. SR (successful rate) indicates the success rate of optimization, and the ideal optimal value of the function is set as R. when the relationship of them satisfies (18), the optimization is successful.

$$|\text{BEST} - R| \leq 1\text{OE} - 6. \tag{18}$$

4.2. Simulation Experiment and Result Analysis. Because different authors adopt various parameters, this paper selects the same general parameters; for example, the population size is 100 (wolves, particle swarm, fish swarm, bee colony, etc.), the maximum number of iterations in the first set of experiments is 200, and the maximum number of iterations in the second set is 2000. The other parameter settings are shown in Table 2.

The above experimental environment is as follows: HP Shadow Genie 4, Windows 10, inter® Core™ i7-8750H; the program is implemented by MATLAB R2017b, M language.

In the above table, $T_{\max}$ means the maximum scouting times. S means step length factor. $\omega\omega$ means determination distance factor. β means renewal scaling factor. ω means inertia weight. c_1 and c_2 mean learning factor. limit_v means individual speed limit. N means number of bees. limit means controls parameter. trynum means maximum number of temptations. v means sense of distance. de means crowding factor. step means step size.

4.2.1. Analysis of the First Set of Experimental Results. Table 3 shows the optimal value, mean value, worst value, standard deviation, and success rate of the seven algorithms for ten standard test functions.

Table 3 shows the comparison of the first set of experimental results of the seven algorithms. The analysis shows that the success rate of CDWPA for ten test functions is 100%, and the results are better than WPA, OWPA, and CWPA, except that the performance of Booth optimization of 2D function is worse than ABC, the performance of other function optimization is better than other algorithms, and the convergence accuracy of CDWPA in $f_2 f_3 f_5 f_6 f_7 f_9, f_{10}$ is improved. In particular, for functions f_1 and f_6, their convergence results achieved the theoretical optimal value 0. Simulation results show the effectiveness of the improved CDWPA.

In the case of the same population size and iteration times, on the solution of the low peak one-dimensional separable function sphere, the success rate of PSO, ABC, and ASFA was 0. The success rate of WPA, OWPA, CWPA, and CDWPA was 100%. Moreover, CDWPA is obviously superior to other algorithms in terms of optimal value, worst value, standard deviation, and average value, which indicates that CDWPA owns better calculation accuracy and robustness.

Except for ABC, all algorithms can successfully search for the solution of the unimodal low-dimensional indivisible functions Easom and Matyas. OWPA has the best optimal value, and CDWPA follows closely with slight difference. Moreover, it is better than OWPA and other algorithms in mean value, standard deviation, and STD, which shows that CDWPA algorithm has stronger stability and balance ability.

When solving the low-dimensional, multipeak, and separable function Booth, the success rate of all algorithms is 100% except ASFA which is 95%. CWPA has the best optimal value. Compared with WPA and the two improved wolf pack algorithms OWPA and CWPA, CWPA is better in terms of mean, variance, and worst value. However, CDWPA has no obvious advantage compared to ABC.

In solving the step of middle dimension, unimodal and separable functions, WPA and CDWPA can find the theoretical optimal value, and the success rate is 100%. CDWPA has fewer iterations and higher optimization efficiency. However, the OWPA with the same reverse strategy holds a success rate of only 50%, and its convergence accuracy is lower than that of CDWPA. In CDWPA, chaotic sequences are used to optimize the initial population.

For solving unimodal and multidimensional inseparable function on Eggcrate, the success rate of PSO, ABC, WPA, and CDWPA is 100%, and that of OWPA and CWPA is 95% and 40%, respectively, yet the success rate of ASFA is only 5%. CDWPA has the best performance in respect of optimal value optimization and is many orders of magnitude more accurate than other algorithms. At the same time, it performs better in mean value, standard deviation, and worst value.

For 200-dimensional functions, PSO, ABC, and ASFA cannot be well optimized when the number of iterations is set to 200, making it easy to fall into local optimization and

Table 1: Benchmark test function.

No.	Name	Expression	Trait	Search range	Min
1	Easom	$f(x) = -\cos(x_1)\cos(x_2) \times e^{(-x_1-\pi)2-(x_2-\pi)^2}$	UN	$[-100, 100]^2$	−1
2	Sphere	$f(x) = \sum_{i=1} x_i^2$	US	$[-1.5, 1.5]^2$	0
3	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	UN	$[-10, 10]^2$	0
4	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	MS	$[-32, 32]^2$	0
5	Eggcrate	$f(x) = x_1^2 + x_2^2 + 25(\sin x_1)^2 + (\sin x_2)^2$	MN	$[-2\pi, 2\pi]^2$	0
6	Step	$f(x) = \sum_{i=1}^{D} \lceil (x_i + 0.5)^2 \rceil$	US	$[100, 100]^{30}$	0
7	Sumsquares	$f(x) = \sum_{i=1}^{D} (x_i)^2$	US	$[-10, 10]^{100}$	0
8	Quadric	$f(x) = \sum_{i=1}^{D} (\sum_{k=1}^{D} x_k)^2$	MS	$[-30, 30]^{100}$	0
9	Griewank	$f(x) = (\sum_{i=1}^{D} (x_i)^2/4000) - (\prod_{i=1}^{D} (x_i^2/\sqrt{x_i}) + 1)$	MN	$[-600, 600]^{100}$	0
10	Ackley	$f(x) = -20e^{-0.2\sqrt{\sum_{i=1}^{D} (x_i^2/D)}} - e^{\sum_{i=1}^{D} \cos((2\pi x_1)/D)} + 20 + e$	MN	$[-32, 32]^{100}$	0

Table 2: Parameter setting of methods.

Methods	Parameter setting
WPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
OWPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
CWPA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
CDPWA	$T_{\max} = 10, S = 0.04, \omega\omega = 0.06, \beta = 5$
PSO	$\omega = 0.7298, c_1 = c_2 = 1.4946, \lim it_v \in (-0.5, 0.5)$
ABC	$N = 2, \lim it = 100$
ASFA	trynum = 100, $v = 1$, de = 0.618, step = 0.1

difficult to escape. WPA, OWPA, CWPA, and CDWPA proposed in this paper have good performance on high-dimensional functions. By analyzing and comparing the characteristics of these algorithms in solving high-dimensional functions, the characteristics of CDWPA are illustrated.

First, the function Sumsquares belongs to the high-dimensional, unimodal, separable function, and Quadric is a peak multidimensional separable function. The success rate of WPA, OWPA, and CDWPA is 100%, and the success rate of CWPA is 90%. Compared with function f_5, the only difference from function f_7 and function f_8 is that both of them are high-dimensional function. This shows that CDWPA has its own advantages in high-dimensional function optimization, and CDWPA has improved a lot compared with the optimal value, average value, variance, and other indicators.

For Griewank and Ackley with 200-dimensional, multimodal, separable function, the success rate of WPA is 95% and 96%, and that of CWPA is 100% and 0, respectively, which are easy to fall into local extremum. For OWPA and CDWPA, the success rate is 100%, while CDWPA can get global extremum. At the same time, by comparing OWPA with CDWPA further, it turns out that CDWPA is better than OWPA in the precision of optimal value, and CDWPA is less than OWPA in variance, which indicates that CDWPA is better than OWPA in robustness.

For low-dimensional functions such as f_1, f_3, f_4, and f_5, the classical intelligent optimization algorithms of PSO, ABC, and ASFA have a good solving effect. Although CDWPA effect is improved, it is not very obvious, and the solving effect is even slightly lower than that of ABC on function f_4. However, as the dimensions rise to 100, the CDWPA advantage becomes apparent. The analysis shows that, for most heuristic optimization algorithms, the generation of random variables is usually based on a certain standard probability distribution, such as uniform distribution and Gaussian distribution. As chaos not only has randomness but also has better spatial ergodicity and unrepeatability, the algorithm is more diverse after adding chaos to random search based on certain probability distribution, and the possibility of jumping out of local extremum point is greater, which enables the algorithm to search at a relatively faster speed. When the dimension is low, the solution space is relatively simple, the distribution of the initial population can meet the needs of the solution space, and the advantages of chaos are not obvious. However, when the solution space becomes more complex, the quality of the solution is required to be higher, and the advantages of chaos and disturbance are fully embodied. This is the reason why the optimization effect of the algorithm is improved significantly in the case of high-dimensional complex functions and is not obvious in the case of low-dimensional functions.

4.2.2. Analysis of the Second Set of Experimental Results. Tables 4 and 5 show the mean value, variance, and optimal value of the four test functions calculated 20 times by seven algorithms on 500 and 1000 dimensions, respectively. Tables 6 and 7 show the P values for the rank sum test. Figures 7 and 8 show the optimization curves of seven algorithms in 500 and 1000 dimensions.

The advantages of CDWPA are not very obvious on low-dimensional functions. However, when the function dimension increases to 500 or even 1000 dimensions, the results of CDWPA are better than other algorithms in optimal value, mean value, and standard deviation. In the case of 500 dimensions, CDWPA achieves the theoretical optimal value of 0 towards Griewank function. With the increase of function dimension, the convergence accuracy of several algorithms decreases, but the optimization effect of CDWPA is obviously better than that of other algorithms

Table 3: The results of the first experiment were compared.

Dim	Fun	Index	PSO	ABC	AFSA	WPA	OWPA	CWPA	CDWPA
		BEST	**−1.00**	**−1.00**	**−1**	**−1**	**−1.00**	**−1.00**	**−1.00**
		MEAN	**−1.00**	**−1**	−0.7254	**−1.00**	**−1.00**	**−1.00**	**−1.00**
2	**Easom** f_1	WORST	**−1.00**	**−1**	$-8.08e-05$	**−1.00**	−0.99	−0.99	**−1.00**
		STD	$5.57e-07$	**0**	0.43	$6.25e-07$	$2.58e-05$	$2.71e-05$	$1.80e-07$
		SR/%	**100%**	**100%**	0	**100%**	35%	65%	**100%**
		BEST	7.88	50.20	444	$6.41e-17$	$4.10e-17$	$2.21e-10$	$\mathbf{1.67e-18}$
		MEAN	10.30	58.06	468.78	$\mathbf{7.17e-16}$	$8.24e-16$	$2.14e-09$	$7.19e-15$
2	**Sphere** f_2	WORST	12.24	66.75	488.93	$6.53e-15$	$2.79e-15$	$1.74e-08$	$\mathbf{4.53e-16}$
		STD	1.36	4.47	12.99	$1.43e-15$	$8.24e-14$	$3.78e-09$	$\mathbf{1.27e-15}$
		SR/%	0	0	0	**100%**	**100%**	**100%**	**100%**
		BEST	$2.53e-10$	$1.47e-06$	$7.29e-10$	$1.50e-17$	$7.67e-26$	$1.22e-15$	$9.78e-25$
		MEAN	$1.16e-08$	$9.91e-05$	$4.08e-7$	$2.46e-07$	$8.38e-20$	$1.66e-7$	$\mathbf{3.46e-20}$
2	**Matyas** f_3	WORST	$5.38e-08$	$4.16e-04$	$7.07e06$	$9.86e-07$	$5.55e-19$	$1.66e-6$	$\mathbf{2.54e-19}$
		STD	$1.56e-10$	$1.06e-04$	$1.57e-06$	$2.51e-07$	$1.52e-19$	$4.17e-07$	$\mathbf{7.50e-20}$
		SR/%	100%	15%	100%	100%	100%	100%	**100%**
		BEST	$7.12e-09$	$\mathbf{1.50\ e-13}$	$1.50e-08$	$1.61e-08$	$1.48e-07$	$1.22e-07$	$1.46e-09$
		MEAN	$3.67e-07$	$\mathbf{4.82\ e-11}$	$3.39e-06$	$3.07e-07$	$2.13e-06$	$2.71e-07$	$1.44e-07$
2	**Booth** f_4	WORST	$1.53e-06$	$\mathbf{2.19\ e-10}$	$2.06e-05$	$2.53e-06$	$5.80e-06$	$1.30e-06$	$6.25e-07$
		STD	$3.57e-07$	$\mathbf{5.66\ e-11}$	$4.65e-06$	$5.92e-07$	$1.75e-06$	$3.74e-07$	$1.58e-07$
		SR/%	**100%**	**100%**	95%	**100%**	**100%**	**100%**	**100%**
		BEST	$4.56e-08$	$7.33e-21$	$7.66e-06$	$3.53e-21$	$1.77e-21$	$6.83e-15$	$\mathbf{1.28e-22}$
		MEAN	$4.61e-06$	$2.55e-18$	1.90	$9.25e-19$	$9.31e-17$	$1.37e-04$	$\mathbf{4.21e-20}$
2	**Eggcrate** f_5	WORST	$2.08e-05$	$1.20e-17$	9.48	$6.71e-18$	$1.05e-15$	0.001	$\mathbf{3.36e-20}$
		STD	$5.31e-06$	$2.89e-18$	3.89	$1.64e-18$	$2.48e-16$	$2.48e-04$	$\mathbf{8.49e-20}$
		SR/%	**100%**	**100%**	5%	**100%**	**100%**	40%	**100%**
		BEST	0	0	199592	**0**	0	7	**0**
		MEAN	0.43	4.25	$2.39e+05$	**0**	0	34.45	**0**
30	**Step** f_6	WORST	2	16	279999	**0**	4	138	**0**
		STD	0.55	4.01	$1.90e+04$	**0**	1.01	29.22	**0**
		SR/%	60%	5%	0	**100%**	50%	0	**100%**
		BEST	334.29	$6.35e+04$	$1.51e+04$	$2.69e-13$	$1.09e-12$	$8.38e-08$	$\mathbf{1.83e-14}$
		MEAN	402.34	$1.02e+05$	$1.61e+04$	$3.53e-12$	$2.68e-12$	$8.48e-07$	$\mathbf{1.24e-13}$
100	**Sumsquares** f_7	WORST	518.25	$1.17e+05$	$1.73e+04$	$1.47e-11$	$2.49e-11$	$2.27e-06$	$\mathbf{5.40e-13}$
		STD	44.88	$1.21e+04$	588.52	$4.12e-12$	$5.44e-12$	$6.76e-07$	$\mathbf{1.29e-13}$
		SR/%	0	0	0	100%	100%	100%	**100%**
		BEST	11.82	$1.66e+04$	$4.99e-05$	$1.40e-12$	$1.02e-12$	$5.33e-07$	$\mathbf{1.81e-13}$
		MEAN	19.44	$2.42e+04$	0.02	$2.90e-11$	$1.90e-11$	$3.92e-06$	$\mathbf{1.98e-12}$
100	**Quadric** f_8	WORST	30.10	$3.03e+04$	0.18	$1.38e-10$	$9.85e-11$	$2.04e-05$	$\mathbf{1.82e-11}$
		STD	4.37	$3.70e+03$	0.05	$3.88e-11$	$2.80e-11$	$4.59e-06$	$\mathbf{4.03e-13}$
		SR/%	0	0	0	100%	100%	90%	**100%**
		BEST	0.04	298	$8.59e+03$	$1.84e-13$	$1.31e-13$	$5.52e-09$	$\mathbf{8.50e-14}$
		MEAN	0.06	426.21	$9.32e+03$	0.04	$1.59e-12$	$2.34e-08$	$\mathbf{1.72e-13}$
100	**Griewank** f_9	WORST	0.09	513.55	$1.00e+04$	0.82	$7.11e-12$	$7.30e-08$	$\mathbf{1.00e-11}$
		STD	0.01	60.07	381	0.18	$2.37e-12$	$1.90e-08$	$\mathbf{2.64e-13}$
		SR/%	0	0	0	95%	100%	100%	**100%**
		BEST	2.23	19.70	$1.59e+16$	$6.94e-08$	$6.45e-08$	$3.62e-05$	$\mathbf{3.68e-09}$
		MEAN	2.50	19.91	$8.01e+14$	$2.12e-07$	$1.43e-07$	1.51	$\mathbf{1.61e-09}$
100	**Ackley** f_{10}	WORST	2.73	20.04	$8.88e+08$	$6.28e-07$	$6.20e-07$	3.59	$\mathbf{3.76e-09}$
		STD	0.11	0.07	$3.56e+15$	$1.51e-07$	$1.46e-07$	1.44	$\mathbf{9.36e-09}$
		SR/%	0	0	0	96%	100%	0	**100%**

and has better advantages in standard deviation, which demonstrates that CDWPA is better in robustness. In general, CDWPA performs better than the other six algorithms in 500- and 1000-dimensional test functions, which verifies the effectiveness of CDWPA in solving high-dimensional complex functions.

For low-dimensional space optimization, the guidance information of optimization is easy to obtain and confirm. The introduction of the optimization strategy based on the idea of "certainty" can improve the algorithm's optimization precision and convergence speed. However, high-dimensional data space to present a kind of highly nonlinear and asymmetric, nonconvex, multipeak complex features, such as value orientation information, is very difficult to obtain accurately; at this point in the optimization strategy, parameter is set to introduce some "random" strategy and will

Table 4: 500-dimensional comparison of experimental results.

Fun		PSO	ABC	ASFA	WPA	OWPA	CWPA	CDWPA
	MEAN	22.86	6.61	$1.36e+03$	$4.61e-43$	$1.09e-08$	$5.56e-89$	$\mathbf{9.53e-149}$
Sphere	STD	5.52	1.28	$1.38e+03$	$1.61e-42$	$1.23e-08$	$1.19e-88$	$\mathbf{2.92e-148}$
	BEST	15.61	4.87	$1.66e+03$	$8.36e-76$	$7.15e-10$	$6.48e-93$	$\mathbf{6.13e-165}$
	MEAN	$4.16e+03$	$6.29e+04$	$5.66e+04$	$1.03e-39$	$8.20e-05$	$1.65e-85$	$\mathbf{1.11e-144}$
Sumsquares	STD	287.36	$2.47e+04$	907.03	$3.35e-39$	$7.39e-05$	$5.63e-85$	$\mathbf{3.21e-144}$
	BEST	$3.48e+03$	$1.75e+04$	$5.43e+04$	$5.75e-72$	$1.00e-05$	$1.53e-88$	$\mathbf{1.92e-160}$
	MEAN	0.11	426	$5.39e+04$	0.05	0.0015	$8.88e-17$	$\mathbf{1.92e-160}$
Griewank	STD	0.009	60.07	$1.01e+03$	0.22	0.0068	$1.17e-16$	**0**
	BEST	0.10	117.15	$5.21e+04$	0	$1.77e-16$	0	**0**
	MEAN	1.71	19.91	$-5.43e+29$	$1.59e-15$	10.32	0.94	**0**
Ackley	STD	0.16	0.078	$2.42e+30$	$1.45e-15$	5.59	1.33	$\mathbf{8.88e-16}$
	BEST	1.43	13.50	$-1.08e+31$	$8.88e-16$	3.30	$7.99e-15$	$\mathbf{8.88e-16}$

Table 5: 1000-dimensional comparison of experimental results.

Fun		PSO	ABC	ASFA	WPA	OWPA	CWPA	CDWPA
	MEAN	74.39	134.78	$2.90e+03$	$3.87e-41$	$5.17e-06$	$1.54e-88$	$\mathbf{3.78e-149}$
Sphere	STD	10.38	6.14	39.26	$1.67e-40$	$4.63e-06$	$3.54e-88$	$\mathbf{1.69e-148}$
	BEST	47.79	125.56	$2.83e+03$	$1.65e-74$	$1.01e-16$	$2.71e-92$	$\mathbf{6.23e-164}$
	MEAN	$3.87e+04$	$2.96e+06$	$1.19e+05$	$2.58e-39$	0.0877	$3.68e-83$	$\mathbf{1.37e-145}$
Sumsquares	STD	$2.16e+03$	$1.59e+05$	$1.69e+03$	$7.68e-39$	0.0707	$3.68e-82$	$\mathbf{6.08e-145}$
	BEST	$3.32e+04$	$2.36e+06$	$1.16e+05$	$1.73e-71$	0.0133	$3.85e-89$	$\mathbf{4.00e-159}$
	MEAN	0.22	$5.6e+03$	$1.11e+05$	0	0.01	$1.55e-16$	**0**
Griewank	STD	0.01	329.22	$1.22e+03$	0	0.03	$1.54e-16$	**0**
	BEST	0.19	$4.89e+03$	$1.09e+05$	0	$1.27e-05$	0	**0**
	MEAN	2.78	19.07	$-7.01e+43$	$2.48e-15$	9.68	0.90	$\mathbf{8.88e-16}$
Ackley	STD	0.06	0.13	$3.08e+44$	$1.81e-15$	4.61	1.29	$\mathbf{8.88e-16}$
	BEST	2.64	18.63	$-1.3e+45$	$8.88e-16$	4.73	$4.44e-15$	**0**

Table 6: P value of different reference functions in 500 dimensions by rank sum test.

		PSO	ABC	ASFA	WPA	OWPA	CWPA
$F1$	P	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$
$F2$	P	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$5.2269e-07$	$3.3819e-04$
$F3$	P	$1.7313e-12$	$8.0065e-09$	$8.0065e-09$	0.3421	$8.0065e-09$	$3.8459e-04$
$F4$	P	$2.3837e-07$	$1.1267e-08$	$1.1267e-08$	$7.10e-03$	$1.1267e-08$	$1.0324e-07$

Table 7: P value of different reference functions in 1000 dimensions by rank sum test.

		PSO	ABC	ASFA	WPA	OWPA	CWPA
$F1$	P	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$
$F2$	P	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$6.7956e-08$	$1.2009e-06$	$2.5960e-05$
$F3$	P	$8.0065e-09$	$8.0065e-09$	$8.0065e-09$	0.3421	$8.0065e-09$	$8.7603e-06$
$F4$	P	$1.1267e-08$	$1.1267e-08$	$1.1267e-08$	$7.10e-03$	$1.1267e-08$	$1.0468e-07$

help the algorithm to maintain the diversity of population, to produce more high-quality data processing, prompting algorithm to jump out of local extremum, and balance the exploration and development ability of the algorithm. Precisely, because the strategy adopted by CDWPA is the "directed random" method, the performance of the low-dimensional solution space is improved, but the gap with other algorithms is not obvious. However, in the high-dimensional solution space, the advantage of "directed random" is clearly reflected.

In order to determine whether the proposed CDWPA is significantly improved compared with other algorithms, this paper carried out nonparametric statistical test, namely, Wilcoxon's rank sum test [33]. The results of CDWPA in each benchmark function of 500 and 1000 dimensions were tested and compared with other algorithms by 5%. Tables 6 and 7 list the P values obtained through the tests, in which P values less than 0.05 indicated that the null hypothesis was rejected, so there was a significant difference at the level of 5%. Conversely, an underlined P value (greater than 0.05)

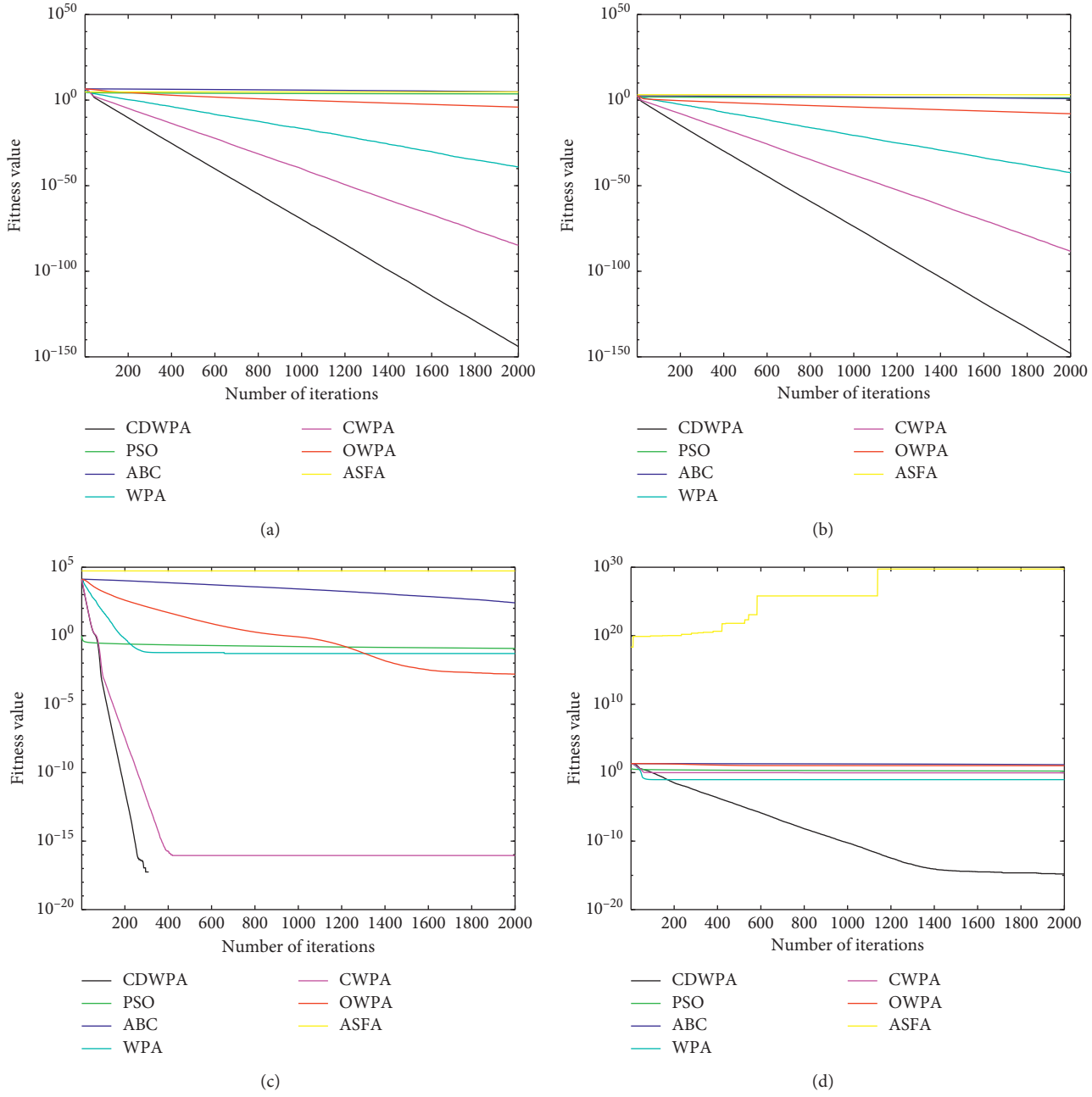


Figure 7: Four kinds of functions find the optimal curve comparison in 500 dimensions.

indicates no significant difference between the comparison values. It can be concluded from the results in Tables 6 and 7 that, in most comparisons, the P values are less than 0.05, which proves that the improvements achieved by the proposed CDWPA are statistically significant for most of the benchmark functions.

Analysis suggests that, for low-dimensional space optimization, the guidance information of optimization is easy to obtain and confirm. The introduction of the optimization strategy based on the idea of "certainty" can improve the algorithm's optimization precision and convergence speed. However, high-dimensional data space to present a kind of highly nonlinear and asymmetric, nonconvex, multipeak complex features, such as value orientation information, is very difficult to obtain accurately; at this point in the optimization strategy, parameter is set to introduce some "random" strategy and will help the algorithm to maintain the diversity of population, to produce more high-quality data processing, prompting algorithm to jump out of local extremum, and balance the exploration and development ability of the algorithm. Precisely, because the strategy adopted by CDWPA is the "directed random" method, the performance of the low-dimensional solution space is improved, but the gap with other algorithms is not obvious. However, in the high-dimensional solution space, the advantage of "directed random" is clearly reflected.

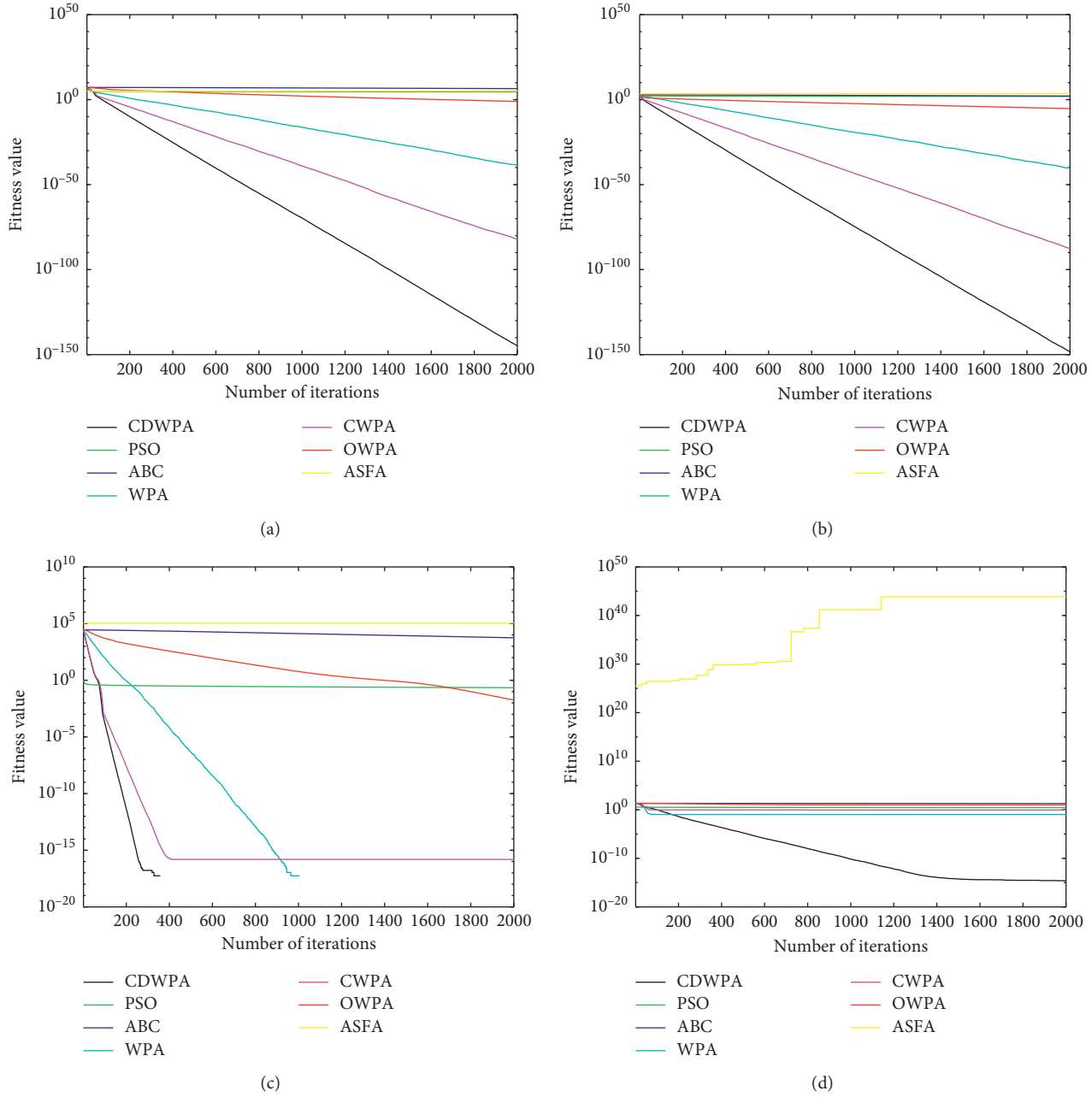


Figure 8: Four kinds of functions find the optimal curve comparison in 1000 dimensions.

Figure 9 shows the comparison of standard deviations of the seven algorithms in optimizing Ackley function in the dimensions of 300–1000. STD reflects the stability of algorithm in the optimization process, with dimension rising and solution space becoming more and more difficult. From the analysis of the STD of seven kinds of algorithms in Ackley function, we found that the standard deviation of stable algorithms such as PSO, ABC, OWPA, and CWPA optimization effect is very poor. PSO, ABC, and OWPA can even not succeed in finding the optimal value, were trapped into local optimum, and cannot get out of the local optimal trap after optimal "trap." However, because it falls into the local optimum at dimension 300 and cannot jump out of the local extreme value effectively, it still cannot find the optimization successfully. WPA and CDWPA have a good optimization effect. Interestingly, the change curves of standard deviations of the two algorithms are similar, with large pulsations at 500 and 1000 dimensions, indicating that the algorithm can effectively jump out of the local optimum and find better targets when it conducts optimization in higher dimensions. OWPA adopts the reverse strategy to optimize the initial population, but it does not consider the distribution location of the initial population, which easily leads to the aggregation of the population. The initial populations of WPA and CWPA were relatively dispersed, but due to the low quality of the initial populations and the

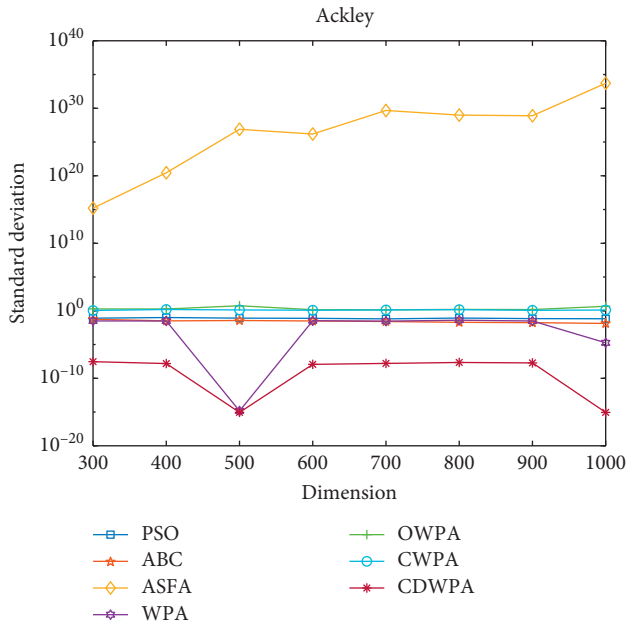


Figure 9: Comparison STD of Ackley with 7 algorithms in 300–1000 dimensions.

relatively fixed migration direction, the convergence accuracy was not high in the optimization. However, CDWPA considers both the initial population distribution position and the initial population mass and adds disturbance factors to the movement of wolves, which equips CDWPA with the most efficiency in high-dimensional function optimization.

5. Conclusion

In order to improve the convergence accuracy of wolf pack algorithm (WPA) in solving high-dimensional complex functions, it is easy to fall into local optimization. In this paper, we use logical self-mapping to generate chaotic sequence in the reinitialization stage, and optimize the initial population through the reverse wolf pack, adding disturbance factor in the process of scouting, which improves the ability of wolf detection to jump out of local optimum. Meanwhile, according to the distance away from the lead wolf, the strong wolf adopts appropriate step size to approach the lead wolf. The convergence of the algorithm is tested by Markov process with the selection of different types of functions. The simulation results show that the comprehensive performance of CDWPA is better than that of wolf pack algorithm (WPA), oppositional wolf pack algorithm (OWPA), and chaotic wolf pack algorithm (CWPA), and three widely applied swarm intelligence algorithms (PSO, ABC, and ASFA) are more suitable for solving problems of high-dimensional function optimization. The improved method of the algorithm can be applied to other intelligent optimization algorithms. This algorithm can be applied to large-scale scheduling problem, photovoltaic power generation problem, multidimensional data clustering, and other problems.

However, practical problems are often more complex and changeable, and all methods cannot be perfect. Premature convergence still restricts the development of swarm intelligence algorithm. In the next step, we should focus on solving this problem to further improve the algorithm performance and efficiency. In addition, it is still the next key research direction to expand the application scope of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by military science project of National Social Science Foundation (2019-SKJJ-C-092), National Natural Science Foundation of China (Grant no. 61502534), Natural Science Foundation of Shanxi Province (No. 2020JQ-493), Military Equipment Research Project (WJ20191C080073-17 and WJ2020A020029), and Research Foundation of Armed Police Force Engineering University (Nos. WJY201922 and JLY2020084).

References

[1] X. Wen, "A dynamic scheduling method for logistics tasks oriented to intelligent manufacturing workshop," *Journal of Mathematical Problems in Engineering*, vol. 2019, Article ID 7237459, 18 pages, 2019.

[2] Y. Tang and Y. Huang, "Framework for artificial intelligence analysis in large-scale power grids based on digital simulation," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 4, pp. 459–468, 2018.

[3] L. Yue and M. Hang, "Studies of sub-synchronous oscillations in large-scale wind farm integrated system," *Iop Conference*, vol. 108, Article ID 052075, 2018.

[4] R. Patriarca and S. Hosseini, "Modeling and quantification of resilience in complex engineering systems," *Complexity*, vol. 2019, Article ID 1038908, 2 pages, 2019.

[5] Y. Zhang, "A survey on optimal control and operation of integrated energy systems," *Complexity*, vol. 2019, Article ID 9462518, 14 pages, 2019.

[6] A. Wadood, "Application of the jaya algorithm in solving the problem of the optimal coordination of overcurrent relays in single- and multi-loop distribution systems," *Complexity*, Article ID 5876318, 13 pages, 2019.

[7] N. Ao, M. Zhao, Q. Li, S. Qu, and Z. Wu, "Network characteristics for neighborhood field algorithms," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12061–12078, 2020.

[8] R.-L. Tang, Wu, and Y.-J. Fang, "Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems," *Soft Computing*, vol. 21, no. 16, pp. 4735–4754, 2017.

[9] Q. Weibiao, "Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function

optimization problems," *IEEE Access*, vol. 7, no. 1104, pp. 72–86, 2019.

[10] H. S. Wu, "New swarm intellingence algorithm—wolf pack algorithm," *Systems Engineering and Electronic Technology*, vol. 35, no. 11, pp. 2430–2438, 2013.

[11] J. Hu, "Swarm intelligence-based optimisation algorithms: an overview and future research issues," *International Journal of Automation and Control*, vol. 14, no. 5-6, pp. 656–693, 2020.

[12] J. Kennedy, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference Neural Networks*, vol. 4, no. 8, pp. 1942–1948, Perth, Australia, 2011.

[13] P. Duan and Y. Ai, "Research on an improved ant colony optimization algorithm and its application," *International Journal of Hybrid Information Technology*, vol. 9, no. 4, pp. 223–234, 2016.

[14] J. Li, Q. Pan, and S. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9353–9371, 2012.

[15] L. Chambers, "Practical handbook of genetic algorithms," *Organization for Security and Co-operation in Europe (OSCE)*, vol. 1, no. 491, p. 346, 1995.

[16] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.

[17] C. Gao, "Optimization of hydraulic turbine governor parameters based on WPA," *IOP Conference Series: Earth and Environmental Science*, vol. 108, no. 5, Article ID 052011, 2018.

[18] L.. Hongdan, R. Sun, and Q. Liu, "The tactics of ship collision avoidance based on Quantum-behaved Wolf Pack Algorithm," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 6, 2019.

[19] H. Li, P. Bai, and H. S. Wu, "Hybrid binary Wolf pack algorithm for the 0-1 multidimensional knapsack problem," *International Journal of Wireless and Mobile Computing*, vol. 12, no. 3, pp. 291–304, 2017.

[20] J. Hu, H. Wu, R. Zhan, and N. Li, "Hybrid integer-coded wolf pack algorithm for multiple-type flatcars loading problem," *Rail Transport Planning & Management*, Article ID 100201, 2020.

[21] Z. Xiu and W. Zhen-Hua, "Improved wolf pack algorithm based on tent chaotic mapping and levy flight," in *Proceedings of the 2017 International Conference on Robots & Intelligent System (ICRIS)*, pp. 165–169, IEEE Computer Society, Huaian, China, 2017.

[22] Y. Zhu, "A chaos wolf optimization algorithm with self-adaptive variable step-size," *Aip Advances*, vol. 7, no. 10, Article ID 105024, 2017.

[23] R. H. Wang, X. Li, and N. Li, "Wolf pack algorithm based on Guassian disturbance and chaotic initialization," *Computer Engineering and Design*, vol. 40, no. 10, pp. 2879–2884, 2019.

[24] H. Li and H. Wu, "An oppositional wolf pack algorithm for Parameter identification of the chaotic systems," *Optik*, vol. 127, no. 20, pp. 9853–9864, 2016.

[25] Y. J. Chiang, "Exponentially harmonic maps, Gauss maps and Gauss sections," *Mediterranean Journal of Mathematics*, vol. 16, no. 5, 2019.

[26] H. Gan, S. Xiao, Z. Zhang, S. Shan, and Y. Gao, "Chaotic compressive sampling matrix: where sensing architecture meets sinusoidal iterator," *Circuits, Systems, and Signal Processing*, vol. 39, no. 3, pp. 1581–1602, 2020.

[27] M. Li, X. Zheng, and Y. Zhang, "Detection of subtle differences in seismic amplitude using convergence rate of the logistic map," *Geophysics*, vol. 84, no. 1, 2019.

[28] X. Jun-Jie, "A smart wolf pack algorithm and its convergence analysis," *Control & Decision*, vol. 31, no. 12, pp. 2131–2139, 2016.

[29] D. B. Fogel, "Evolutionary algorithms in theory and practice," *Complexity*, vol. 2, no. 4, pp. 26-27, 1997.

[30] Y. W. Zhang, L. Wang, and Q. D. Wu, "Dynamic adaptation cuckoo search algorithm," *Control and Decision*, vol. 29, no. 4, pp. 617–622, 2014.

[31] O. Castillo and P. Melin, "Optimization of benchmark mathematical functions using the firefly algorithm with dynamic parameters," *Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics*, vol. 574, no. 1, pp. 81–89, 2015.

[32] Y. Baba and O. C. Ugweje, "Development and analysis of a modified artificial fish swarm algorithm," in *Proceedings of the 2017 13th International Conference on Electronics*, pp. 1–6, Computer and Computation(ICECCO), Abuja, Nigeria, 2017.

[33] R.. Wang and R. C. Purshouse, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2012.

Hindawi
Complexity
Volume 2021, Article ID 6667516, 16 pages
https://doi.org/10.1155/2021/6667516

WILEY | Hindawi

Research Article

A Novel MILP Model for the Production, Lot Sizing, and Scheduling of Automotive Plastic Components on Parallel Flexible Injection Machines with Setup Common Operators

Beatriz Andres, Eduardo Guzman, and Raul Poler

Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València (UPV), Calle Alarcón, 03801 Alcoy, Alicante, Spain

Correspondence should be addressed to Beatriz Andres; bandres@cigip.upv.es

Received 6 November 2020; Revised 21 January 2021; Accepted 28 January 2021; Published 11 February 2021

Academic Editor: Shangce Gao

In this article, a mixed integer linear program (MILP) model is proposed for the production, lot sizing, and scheduling of automotive plastic components to minimize the setup, inventory, stockout, and backorder costs, by taking into account injection molds as the main index to schedule on parallel flexible injection machines. The proposed MILP considers the minimum and maximum inventory capacities and penalizes stockout. A relevant characteristic of the modeled problem is the dependence between mold setups to produce plastic components. The lot sizing and scheduling problem solution results in the assignment of molds to machines during a specific time period and in the calculation of the number of components to be produced, which is often called lot size, following a sequence-dependent setup time. Depending on the machine on which the mold is setup, the number of units to be produced will be distinct because machines differ from one another. The stock coverage, defined in demand days, is also included in the MILP to avoid backorders, which is highly penalized in the automotive supply chain. Added to this, the proposed model is extended by considering setup common operators to respond to and fulfill the constraints that appear in automotive plastic enterprises. In this regard, the MILP presented solves a lot-sizing and scheduling problem, emerged in a second-tier supplier of a real automotive supply chain. Finally, this article validates the MILP by performing experiments with different sized instances, including small, medium, and large. The large-sized dataset is characterized by replicating the amount of data used in the real enterprise, which is the object of this study. The goodness of the model is evaluated with the computational time and the deviation of the obtained results as regards to the optimal solution.

1. Introduction

Production planning, sequencing, and scheduling are key operations performed by enterprises, and any circumstances or events that affect them strongly influence the supply chain operation in which they are embedded. All these three planning levels are characterized by the decision-making time horizon in accordance with three decision-making levels: strategical, tactical, and operational. Thus, production planning is set at the strategic level by considering families of products, while sequencing and scheduling are set at a more operational decision-making level. Accordingly, Gujjula et al. [1] proposes the following differentiation between these two concepts: (i) production scheduling, which deals with the assignment of production orders to production intervals with a short planning horizon and specific time periods and (ii) production sequencing, which deals with the sequence of production orders for each production interval.

Our aim is to solve a lot-sizing and scheduling problem with a sequence-dependent setup on parallel flexible machines. To this end, a mixed integer linear program (MILP) model is proposed to minimize the setup, inventory, stockout, and backorder costs by taking into account injection molds as the main index to schedule on parallel flexible injection machines. We also consider setup common operators to extend the proposed basic MILP model.

This article focuses on the specific production lot-sizing and scheduling problem (LSSP) in parallel flexible machines.

According to Kim et al. [2], the LSSP deals with the minimization of production and inventory costs by simultaneously optimizing lot sizes and production schedule. The author also refers to small and big bucket models depending on the number of allowed setups. Accordingly, production scheduling assigns production orders to production intervals with a short-term planning horizon lasting several days or shifts. Moreover, the assignment in the scheduling process has an implicit sequence for each production shift. Therefore, the main goal is to identify the time period when to produce, the quantity to produce as the lot size in units or timeslots, and the production sequence required to meet demand and to avoid backorders and stockouts.

As enterprises are seen as complex systems in operations management, their operation mechanisms are difficult to manage. This difficulty further increases because enterprises belong to a supply network system, in which the complexity and relationships with external actors are latent. Current global market conditions and constant changes in the supply chain environment render enterprises as complex systems. Moreover, researchers have to bear in mind that market consumers are currently used to acquire highly personalized products, which is known as mass customization with short development periods ("time to market").

In the last few years, novel technologies have been increasingly used, such as cloud computing, big data, artificial intelligence, and machine learning. This has become a trend that has boosted companies to transform their way of operating at enterprise and supply chain levels. The result of this digital transformation has been coined as Industry 4.0 [3]. Nevertheless, the application of Industry 4.0 to small- and medium-sized enterprises (SMEs) is not as idealistic as the Industry 4.0 definition indicates. The literature includes different studies on implementing Industry 4.0 technologies into different sectors. The present work focuses on the literature review by Echchakoui and Barka [4], which studies the impact that Industry 4.0 has on the plastics industry. The relevance of the work by Echchakoui and Barka [4] is aligned with the present study because we develop MILP to support the LSSP in a company that belongs to the plastic sector by particularly focusing on the injection of plastic automotive components. In the aforementioned literature review, the authors highlight that implementing Industry 4.0 research into the plastics industry is still in its initial stages, but research in this field is growing.

The origin of this article lies in the H2020 Project Cloud Collaborative Manufacturing Networks (C2NET) [5], whose research focuses on providing a cloud platform, in which a set of tools and technologies are embedded to support: (i) data management and interoperability; (ii) the optimization of plans at enterprise and supply chain levels; and (iii) the integration of collaborative processes among supply network members. The C2NET cloud platform is built according to an open-source philosophy, which makes it affordable and easy to use by SMEs in terms of both monetary and expertise or knowledge required for its use. The optimization module contains advanced optimization models and algorithms to support and calculate replenishment, production, and delivery plans at both the enterprise and network levels. The calculation of plans at the enterprise level is characterized by only considering the resources and data from a single enterprise, while the calculation of plans at the network level uses the constraints, resources, and data from two supply chain enterprises, or more, as input data. When solving replenishment, production, and delivery plans from a collaborative network perspective, the output data of one enterprise plan are used as the input data of another enterprise plan. If we consider two enterprises from a network, A and B, where A is the supplier and B is the manufacturer, the replenishment plan of company B is constrained by the production plan of company A. Hence, information is exchanged and a loop plan is calculated until the materials required by company B coincide with the materials that can be produced by company A. On the C2NET cloud platform (CPL), the negotiation loop is operated by the collaborative module (COT), the plan calculation is managed by the optimization module (OPT), and data exchange and information interoperability are handled by the data collection framework module (DCF) [6]. The operation and management of the complex large-scale systems that characterize enterprises are covered by the models and tools developed in the CLP with the help of embedded intelligent methods.

Different enterprises from diverse sectors were involved in the C2NET project to validate and test the generated results. The enterprise study object of this article is the automotive industry, which is included in one of the industrial pilots. The automotive supply chain is characterized by the need to perform flexible manufacturing to meet the demand of the original equipment manufacturer (OEM) in terms of delivery time, scheduling, and lot size in a just in time (JIT) production system [7]. When addressing the LSSP in the automotive industry, suppliers are highly penalized when the components supplied to the OEM are delayed. Thus, the automotive components' manufacturers have to manage minimum and coverage stocks to avoid potential penalizations. The coverage goal is set in demand days so that suppliers have to produce at least 3 days of demand in advance (commonly used in the automotive industry) to stock production, as advances to avoid delaying demand because it could imply stopping the automotive assembly line in OEM facilities.

The main objective of this article is to model a real problem from a second-tier supplier in the automotive supply chain and to solve it in a reasonable computation time. Accordingly, a lot-sizing and sequencing model is hereafter provided to respond to the requirements of a second-tier supplier from an automotive supply chain. The proposed optimization model is embedded in the OPT module of the C2NET CPL. A novel MILP model for automotive plastic components production lot sizing and scheduling on parallel flexible injection machines is proposed with setup common operators. The MILP model was implemented in Pyomo [8], used as an extensible python-based open-source optimization modeling language for linear programming and nonlinear programming, among others. Finally, a complex large-scale problem is addressed to deal with the scheduling plan of an automotive components' manufacturer.

The reminder of this document is organized as follows. The next section presents a literature review to gain insight into works that have addressed similar type problems and to justify this study's contribution. Section 3 describes and analyzes the problem to be solved. Section 4 presents the novel MILP model for the automotive plastic components production lot sizing and scheduling on parallel flexible injection machines. We take this MILP as the base model. Section 5 puts forward an extension of the base model, which contemplates common setup operators in the proposed LSSP to provide a more realistic perspective to the base model by taking into account the studied plastic components injection industry. Section 6 offers comprehensive numerical experimentation by considering different data sizes for both proposed models, including small, medium, and large datasets. This article is concluded by discussing the findings obtained with the addressed problem.

2. Related Studies

A large amount of research has been conducted into different LSSP characteristics, but very few studies present optimization approaches that combine or integrate LSSP characteristics, that is, lot sizing and scheduling. According to Ríos-Solís et al. [9], the product-part-mold-machine (PPMM) problem and the part-mold-machine (PMM) problem study this combination of approaches and indicate that this problem type is scarcely addressed in the literature. Ríos-Solís et al. [9] define products as final products and parts as components that derive from the bill-of-materials parts explosion. Furthermore, Ríos-Solís et al. [9] classify the PPMM as bilevel capitated LSSP. Accordingly, the first part of the PPMM and PMM lot-sizing and scheduling problems seeks to determine the optimal size of either a lot of products in the PPMM approach or the parts in the PMM approach. Both approaches seek to assign parts or pieces to molds and molds to the machines. The second part is involved in the scheduling that aims to determine the order of processing molds in machines during each planning time period.

Studies such as Ibarra-Rojas et al. [10] address the problem of manufacturing parts that are produced in molds and are mounted on machines. This study proposes an MILP that seeks to maximize the production of parts and also determines the batch size of every part and the assignments of parts to mold and machines. An update of this study is that proposed by Ríos-Solís et al. [9], which aims to determine the lot size of a finished product; that is, it determines the number of finished products to be manufactured, the number of parts to be manufactured, and the assignment of the parts to the mold and the mold to the machine. The model of Ríos-Solís et al. [9] also contemplates the feasibility of scheduling molds in machines during each period. However, in their experimental results, they only handle one period of time periods and propose working with multiple time periods as a future research line. Different approaches have been proposed in the literature to optimally solve scheduling in engineering; in this regards, Li et al. [11] highlight the importance on the use and application of MILP to deal with the aforementioned problem in the engineering research area.

Studies about LSSP are paid more attention by researchers and companies, given their applicability to the real world [12]. In our literature review, we find studies that integrate batch sizing decisions into restrictions for scheduling problems (see Table 1), studies such as that presented by Stadtler [13], in which a combined approach to support a single machine LSSP is proposed. For this purpose, an MILP is formulated to minimize maintenance and inventory setup costs over a planning horizon. Wolosewicz et al. [15] combine production planning and scheduling by proposing an MILP that seeks to determine the lot size for a fixed sequence of operations in the machines, taking into account the times and operating costs. Kim et al. [2] present a combined approach in a MILP that seeks to minimize the sum of the cost of production, installation, and inventories. James and Almada-Lobo [16] present an MILP for the scheduling problem and capacitated lot sizing of a single machine and a parallel machine with sequence-dependent setup times and costs. Meyr and Mann [14] put forward an MILP to simultaneously determine production lot sizes and schedules on nonidentical parallel production lines.

Our model deals with the problem contemplated by Ríos-Solís et al. [9] and Ibarra-Rojas et al. [10]. By continuing with the future research lines indicated by these authors, our study considers many periods when modeling and running experiments to solve the LSSP. The base model herein proposed also bears in mind stock coverage constraints, which are typical in the studied automotive supply chain industry context. The proposed novel MILP also contemplates an objective function based on the assembly line and allows idle times among molds, which is a fundamental characteristic for real cases and has been ignored by former studies. Finally, the base model is extended by offering a scenario that comes closer to reality by considering another index that represents the workers who change molds and, thus, bears in mind the casuistry associated with the LSSP that refers to the setups of the usual operators.

3. Problem Description

The main aim of this article is to propose a novel MILP model for the production scheduling carried out by an automotive plastic components manufacturer that acts as a second-tier supplier in the automotive supply chain. Plastic components are produced in molds that are mounted on parallel flexible injection machines. Injection machines shape plastic pellets into automotive semifinished products to then be assembled on an OEM production line. Nevertheless, for industry, studying car components is treated as the second-tier supplier's end products.

According to Ríos-Solís et al. [9], lot-sizing and scheduling plans of mold-injection enterprises entail decision making to determine (i) the lot size as the amount of components to be produced during a period of time or a number of periods to produce the same component and (ii) the assignment of molds to machines by considering that

Table 1: Literature review of recent lot-sizing and scheduling problems.

Authors	Objectives	Industrial application	Resolution methods	Solver
Kim et al. [2]	Minimize the sum of production, setup, and inventory costs	Zinc refinery	Heuristic algorithm combining a decomposition scheme with a local search procedure	CPLEX
Ibarra-Rojas et al. [10]	Maximize the weighted cost of produced pieces	Automotive, consumer goods, and toys	Decomposition approach	CPLEX 11.2
Stadtler [13]	Minimize the sum of inventory holding and setup costs in the planning interval	Pharmaceutical		Xpress-MP
Ríos-Solís et al. [9]	Maximize the profit of finished products	Automotive, consumer goods, and toys	Iterative heuristic based on mathematical programming	Gurobi 6.05
Meyr and Mann [14]	Minimize inventory holding, sequence-dependent setup, and line-specific production costs of potentially heterogeneous production lines	—	Decomposition-based approach	GLPK 4.44
Wolosewicz et al. [15]	Minimize the sum of production, inventory, and setup costs	—	Lagrangian heuristic	XPRESS-MP

components can be produced only in specific molds with their shape cavity.

The second-tier supplier herein studied is characterized by having specific molds that produce each automotive component. When two molds are available to produce the same component, these molds involve different processing times, given their technical particularities. Each mold can also be setup on different machines to produce the same automotive component, but the same mold mounted on different machines has different production rates depending on the machine on which it is setup. Accordingly, productivity differs depending on the mold and machine assignment (see Figure 1). Moreover in the automotive sector, two different parts are produced in the same mold, including right- and left-hand parts, which is known as biproduct injection molding [17].

The studied enterprise has 21 injection machines and a set of molds to produce the range of components delivered to the first-tier supplier and finally to the various OEMs that belong to different automotive supply chains, depending on the car brand. The enterprise works three shifts per day over a 5-day week and reserves overtime shifts on the sixth weekday in the event of production not finishing during the normal operation time. Workers are told that they will work overtime periods at least 1 week beforehand.

Molds are changed using cranes and, therefore, a setup time is incurred. A limited number of workers are in charge of changing and mounting molds to the assigned machines. Moreover, these workers, known as setup common operators, work only two of the three shifts that the enterprise arranges.

One of the requirements of the enterprise's study is that once the mold is setup on a machine, the mold should remain for at least 24 h so as not to saturate the work of the limited setup common operators and to not incur on too many setups because the setup time is estimated between 1 and 3 h, which obviously has an associated setup cost. If a longer production time is needed, the mold is set up during the required time periods without incurring any setup costs. Thus, the modeled problem should contemplate no setup carry-over cases.

Backorders are highly penalized in the automotive supply chain. Therefore, enterprises in the automotive industry use stock coverage, which indicates the number of demand days that stocks can cover. Normally in the automotive industry, stock coverage is set at 3 demand days; for example, for the demand of four units for the next three periods ($d1 = 4$ units, $d2 = 4$ units, $d3 = 4$ units), stock coverage is defined as 12 units at the end of the first period. Moreover, as warehouses have space limitations, a maximum inventory is considered.

With regards to the demand, the automotive industry updates the demand during each period for the next five frozen periods. The OEM works with considerable demand information (1 year of demand horizon), but only communicates the demand for the next 6 months to the first-tier supplier. Finally, first- and second-tier suppliers normally work to a 3-month demand horizon and with daily periods. The LSSP considers a 21-day horizon.

A detailed flowchart of the processes is presented (see Figure 2), from the generation of the customer demand to the final resolution of the second-tier supplier model. Accordingly, in the automotive supply chain, the OEM generates customer orders, according to the final customer demand. The OEM transforms the customer orders into the master production scheduling (MPS), which, jointly with bill of materials (BOM) and the inventory availability, computes the material requirement plan (MRP). The OEM demand plan is generated from the MRP and transferred to the first-tier supplier. Then, the first-tier supplier obtains its demand by using a simple bill-of-materials parts explosion, given the OEM demand plan. The second-tier supplier estimates the component requirements from the final requirements in a frozen sequence transferred by the first-tier supplier, and the demand plan is generated [18]. Then, the second-tier supplier proceeds to compute the LSSP model with the aim of (i) assigning molds to machines; (ii) scheduling the processing molds in machines during each planning time period; and (iii) calculating the optimal lot size of products. The LSSP is modeled by considering molds to be the main index. The

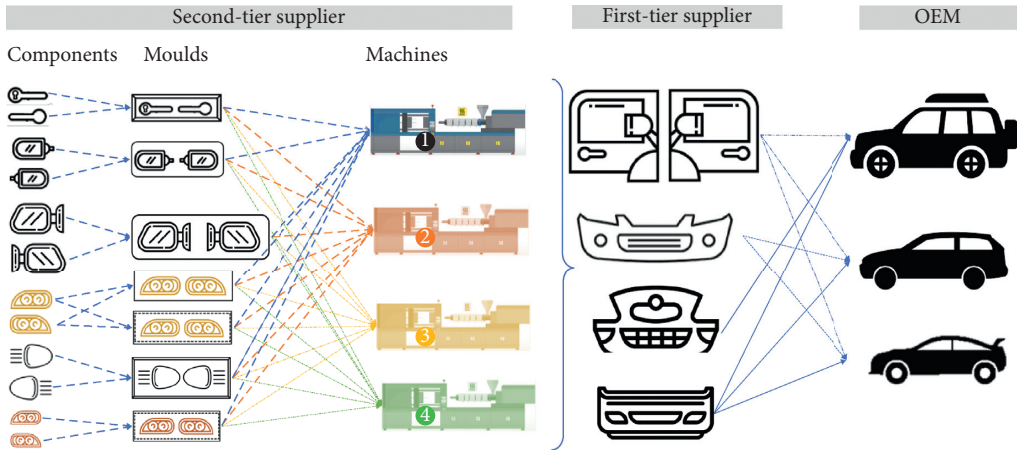


Figure 1: Production scheme on parallel flexible injection machines of plastic components in the automotive supply chain.

proposed production lot-sizing and scheduling base model has as main inputs the data parameters described in Table 2. Nevertheless, if setup common operators are considered when solving the LSSP, data parameters related with the type of operators have to be taken into account for building the extended version of the base model. Finally, the proposed MILP is solved by the second-tier supplier using the Guroby solver obtaining optimal or near optimal solutions of the production LSSP. In a nutshell, the objective of the proposed MILP is to minimize the total costs, namely, setup and inventory costs, penalization costs for coverage stockouts, backorder, and tool setup costs, such as the route cost of selecting one machine or another to setup a tool.

4. MILP Model for Lot Sizing and Scheduling on Parallel Flexible Injection Machines: Notation and Model Formulation

The lot sizing and scheduling on the parallel flexible injection machines problem under study is notated in Table 2, where the main indexes related to machines, tools, products, and periods are notated. The input data parameters are presented, and decision variables are established as the output data of the MILP model for lot sizing and scheduling on parallel flexible injection machines.

The formulation of the MILP model for lot sizing and scheduling on parallel flexible injection machines is described next. The objective function minimizes total costs, including setup and inventory costs, penalization costs for coverage stockouts, backorder, and tool setup costs, such as the route cost of selecting one machine or another to set up a tool.

$$\begin{aligned}\text{Min } z = &\sum_i \sum_j \sum_t cs_j \cdot \text{SA}_{ijt} + \sum_k \sum_t ci_k \cdot \text{INV}_{kt} \\ &+ \sum_k \sum_t cst_k \cdot \text{ST}_{kt} + \sum_k \sum_t cb_k \cdot B_{it} \\ &+ \sum_i \sum_j \sum_t r_{ij} \cdot cr_{ij} \cdot \text{SA}_{ijt}.\end{aligned} \tag{1}$$

It is subject to

Sequence constraints

$$S_{ijt} \le r_{ij}, \quad \forall i, j, t, \tag{2}$$

$$\text{SA}_{ijt} \le r_{ij}, \quad \forall i, j, t, \tag{3}$$

$$\sum_j S_{ijt} \cdot r_{ij} \le 1, \quad \forall i, t, \tag{4}$$

$$\sum_i S_{ijt} \cdot r_{ij} \le a_j, \quad \forall j, t. \tag{5}$$

Constraints (2) and (3) constrain only setup tools j on previously assigned specific machines i. Constraint (4) determines that one or any tool j can be set up for production during each time period t. Constraint (5) guarantees that the total amount of tools j available for production can only be set up as a maximum during each time period t.

4.1. Production and Capacity Constraints.

$$X_{kt} <= \sum_i \sum_j p_{kt} \cdot r_{ij} \cdot \text{TP}_{ijt}, \quad \forall k, t, \tag{6}$$

$$Xnp_{kt} <= \sum_j \sum_i np_{jk} \cdot r_{ij} \cdot \text{SA}_{ijk}, \quad \forall k, t, \tag{7}$$

$$X'_{kt} = X_{kt} - Xnp_{kt}, \quad \forall k, t, \tag{8}$$

$$\text{TP}_{ijt} = tp_t \cdot S_{ijt}, \quad \forall i, j, t. \tag{9}$$

Constraint (6) computes the amount of product k produced during time period t and ensures that a specific tool j is able to be set up on machine i during time period t when product k is produced. Constraint (7) determines the amount of products k no longer produced when tool j is set up during time period t on machine i by considering that another tool j is set up on machine i during time period $t-1$. This also ensures that such a specific tool j is able to be set up on machine i during time period t when product k is produced. Constraint (8) computes the amount of product k

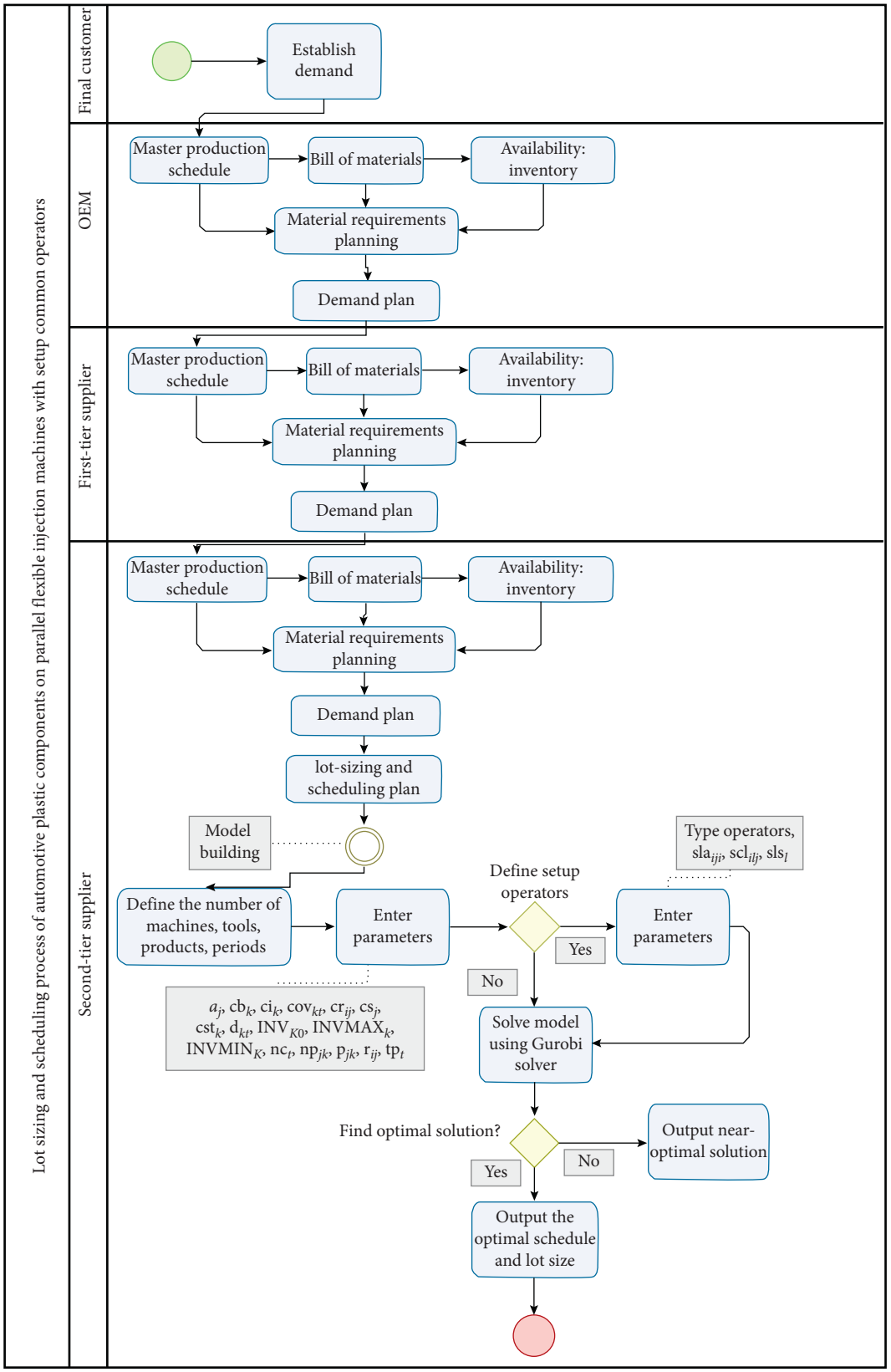


Figure 2: Flowchart of the proposed solution methodology.

Table 2: Nomenclature for the model.

Index	
i	Index of machines $i \in \{1, \ldots, I\}$
j	Index of tools $j \in \{1, \ldots, J\}$
k	Index of products (parts) $k \in \{1, \ldots, K\}$
t	Index of time periods $t \in \{1, \ldots, T\}$
Data	
a_j	Total amount of tools j available for production
cb_k	Backorder cost of product k
ci_k	Inventory cost of product k
cov_{kt}	Stock coverage defined as the number of time periods for the stock minimum coverage of product k during time period t
cr_{ij}	Setup cost of tool j on machine i
cs_j	Setup cost of preparing tool j
cst_k	Coverage stockout cost of product k
d_{kt}	Demand of product k during time period t
INV_{k0}	Initial inventory of product k
INVMAX_k	Maximum inventory units for product k during time period t
INVMIN_k	Minimum inventory units for product k during time period t
nc_t	Amount of tool changes allowed during time period t
np_{jk}	Amount of products k no longer produced when tool j is set up
p_{jk}	Amount of products k produced when tool j is set up
r_{ij}	1 if tool j can be set up on machine i; 0 otherwise
tp_t	Production time available during time period t
Decision variables	
B_{kt}	Backorder of product k during time period t
INV_{kt}	Inventory level of product k at the end of time period t
SA_{ijt}	1 if tool j is set up on machine i during time period t and is not set up on machine i during time period $t-1$; 0 if tool j is set up on machine i during time period $t-1$
S_{ijt}	1 if tool j is set up on machine i during time period t; 0 otherwise
$\mathrm{ST}_{kt}t$	Coverage stockout of product k during time period
TP_{ijt}	Production time of tool j set up on machine i during time period t
X_{kt}	Amount of product k to produce during time period t
Xnp_{kt}	Amount of product k no longer produced during time period t, while a tool is set up
X_{kt}'	Amount of product k to produce during time period t by subtracting Xnp_{kt}

to be produced during time period t by subtracting Xnp_{kt} as the amount of products k no longer produced when tool j is set up. Constraint (9) determines the production time used during time period t when tool j is set up on machine i. In TP_{ijt}, we indicate the production lot size during period times, e.g., 24 h. This means that if tool j is set up on machine i, the tool cannot be changed by another one for the next 24 h. Therefore, the minimum lot size corresponds to the products produced during the 24 h that tool j is set up on machine i.

4.2. Setup Constraints.

$$\mathrm{SA}_{ijt} = S_{ijt}, \quad \forall i, j, t = 1, \tag{10}$$

$$\begin{aligned} &\mathrm{SA}_{ijt} \geq S_{ijt} - S_{ijt-1}, \quad \forall i, j, t > 1, \\ &\mathrm{SA}_{ijt} \leq 1, \quad \forall i, j, t > 1, \end{aligned} \tag{11}$$

$$\sum_i \sum_j \mathrm{SA}_{ijt} \leq nc, \quad \forall t. \tag{12}$$

Constraint (10) allows the first setup of tool j to be determined on machine i, which enables it to be modeled if tool j is set up during time period t on machine i for the first time, and decision variables S_{ijt} and SA_{ijt} take the same value 1. Constraint (11) ensures that SA_{ijt} does not take values above 1. Constraint (12) limits the amount of tool changes allowed during time period t.

4.3. Inventory Balance Equations.

$$\mathrm{INV}_{kt} = \mathrm{INV}_{i0} + X_{kt}' - d_{kt} + B_{kt}, \quad \forall k, t = 1, \tag{13a}$$

$$\mathrm{INV}_{kt} = \mathrm{INV}_{kt-1} + X_{kt}' - d_{kt} + B_{kt} - B_{kt-1}, \quad \forall k, t > 1. \tag{13b}$$

Inventory balance equations (13a) and (13b) guarantee appropriate values for the inventories, quantities to produce, and backorders for each time period $t = 1$ and $t > 1$, respectively.

4.4. Stock Coverage Constraint.

$$\mathrm{INV}_{kt} \geq \mathrm{INVMIN}_{kt}, \quad \forall k, t, \tag{14}$$

$$\mathrm{INV}_{kt} \leq \mathrm{INVMAX}_{kt}, \quad \forall k, t, \tag{15}$$

$$\mathrm{INV}_{kt} + \mathrm{ST}_{kt} \geq \sum_{c=1}^{c=\mathrm{cov}} d_{k(t+c)}, \quad \forall k, t < T - \mathrm{cov}. \tag{16}$$

Constraints (14) and (15) limit the inventory levels for each product k according to the available space for inventory holding during time period t. Constraint (16) is a constraint for the stock coverage of products.

4.5. Bound and Nature Variables.

$$\mathrm{SA}_{ijt}, S_{ijt} \in \{0, 1\}, \quad \forall i, j, t, \tag{17}$$

$$X_{kt}, \mathrm{INV}_{kt}, B_{kt}, \mathrm{ST}_{kt}, \mathrm{Xnp}_{kt}, \mathrm{BX}'_{kt} \in \mathbb{N}, \quad \forall k, t, \tag{18}$$

$$\mathrm{TP}_{ijt} \in \mathbb{N}, \quad \forall i, j, t. \tag{19}$$

Constraint (17) indicates the binary nature of setup S_{ijt} and the setup amount SA_{ijt} variables. Constraints (18) and (19) indicate the continuous nature of the represented variables.

5. MILP Model for Lot Sizing and Scheduling on Parallel Flexible Injection Machines with Setup Common Operators: Notation and Model Formulation

The MILP model for lot sizing and scheduling on parallel flexible injection machines described in the previous section provides a solution to an LSSP that emerged in a second-tier supplier of a real automotive supply chain. In this section, the base model is extended by considering setup common operators to respond and fulfill the constraints that arise in automotive plastic enterprises. In this regard, the base model is extended to offer coming closer to reality by considering another index which represents the workers who change molds. It also bears in mind the casuistry associated with the LSSP problem that refers to setup common operators.

The nomenclature for the extended model is represented in Table 3. In order to avoid repetitions, the data and decision variables used for the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators are considered to be the same as in the base model. Table 3 only shows the new data and decision variables in relation to the previous base model. There are different types of operators and a distinct number of workers corresponding to each operator type. For example, let us consider only one setup operator type that corresponds to the qualified technician category and is specialized in changing molds. Companies have a setup operator type corresponding to the auxiliary technician category, whose task involves helping the qualified technician, as well as a setup operator type corresponding to the mechanic category, and provides support whenever failure of a mechanic, electric, or physical, among others, occurs. All these categories or operator types have varying numbers of workers who go on different shifts. For example, in a company like that herein studied that has 21 machines, there could be two qualified technicians, four auxiliary technicians, and one mechanical technician per shift.

The formulation of the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators is described next. The objective function minimizes total costs, which include setup costs with common setup operators, inventory costs, penalization costs for coverage stockouts, and backorder and tool setup costs, such as the route cost of selecting one machine or another to set up a tool.

$$\begin{aligned}\mathrm{Min}\, z = {} & \sum_i \sum_l \sum_j \sum_t cs_j \cdot \mathrm{SA}_{iljt} + \sum_i \sum_l \sum_j \sum_t slc_{iljt} \cdot \mathrm{SA}_{iljt} \\ & + \sum_k \sum_t ci_k \cdot \mathrm{INV}_{kt} + \sum_k \sum_t cst_k \cdot \mathrm{ST}_{kt} \\ & + \sum_k \sum_t cb_k \cdot B_{it} + \sum_i \sum_l \sum_j \sum_t r_{ij} \cdot cr_{ij} \cdot \mathrm{SA}_{iljt}.\end{aligned} \tag{20}$$

It is subject to the following.

5.1. Sequence Constraints.

$$S_{iljt} \leq r_{ij}, \quad \forall i, l, j, t, \tag{21}$$

$$\mathrm{SA}_{iljt} \leq r_{ij}, \quad \forall i, l, j, t, \tag{22}$$

$$\sum_j S_{iljt} \cdot r_{ij} \leq 1, \quad \forall i, l, t, \tag{23}$$

$$\sum_i S_{iljt} \cdot r_{ij} \leq a_j, \quad \forall j, t, l. \tag{24}$$

Equations (21) and (22) constrain only setup tools j by setup operators l on specific previously assigned machines m. Constraint (23) determines that one or any tool j can be set up by setup operator l for production during each time period t. Constraint (24) guarantees that the total amount of tools j available can only be set up for production as a maximum during each time period t by setup operator l.

5.2. Production and Capacity Constraints.

$$Xnp_{kt} <= \sum_i \sum_l \sum_j np_{jk} \cdot r_{ij} \cdot \mathrm{SA}_{iljk}, \quad \forall k, t, \tag{25}$$

$$\mathrm{TP}_{ijt} = tp_t \cdot S_{iljt}, \quad \forall i, l, j, t. \tag{26}$$

Constraint (25) determines the amount of products k no longer produced when tool j is set up by operator l on machine I during time period t by considering that another tool j is set up on machine i during time period $t-1$. It also ensures that such a specific tool j can be set up on machine i during time period t when product k is produced. Constraint (26) determines the production time spent during time period t when tool j is set up by operator l on machine i.

Table 3: Nomenclature for the model.

Index	
l	Index setup-type operators $l \in \{1, \ldots, L\}$
Data	
sla_{ilj}	Number of setup-type operators l required to setup the tool j on machine i
scl_{ilj}	Cost of type operator l to setup the tool j on machine i
sls_l	Number of available workers of type operator l available
Decision variables	
S_{iljt}	1 if the tool j is setup by setup operator l on machine i during time period t; 0 otherwise
SA_{iljt}	1 if tool j is set up by setup operator l on machine i during time period t and is not set up on machine i during time period $t-1$; 0 if tool j is set up by setup operator l on machine i during time period $t-1$

5.3. Setup Constraints.

$$\text{SA}_{il\,jt} = S_{il\,jt}, \quad \forall i, l, j, t = 1, \tag{27}$$

$$\begin{aligned} &\text{SA}_{il\,jt} \geq S_{il\,jt} - S_{il\,jt-1}, \quad \forall i, l, j, t > 1, \\ &\text{SA}_{il\,jt} \leq 1, \quad \forall i, l, j, t > 1, \end{aligned} \tag{28}$$

$$\sum_i \sum_j \text{SA}_{il\,jt} \leq nc_t, \quad \forall l, t. \tag{29}$$

Constraint (27) allows the first set up of tool j performed by operator l on machine i to be determined and enables it to be modeled if tool j is set up during time period t on machine i for the first time. Decision variables S_{iljt} and SA_{iljt} take the same value 1. Constraint (28) ensures that SA_{iljt} does not take values above 1. Constraint (29) limits the amount of tool j changes allowed during time period t and set up by operator l on machine i.

5.4. Labour Constraint.

$$\sum_i \sum_j \text{SA}_{il\,jt} \cdot sla_{il\,j} \leq sls_l, \quad \forall l, t. \tag{30}$$

Constraint (30) limits the amount of tools changes allowed during time period t to the available number of workers of type operator l by considering the number of setup-type operators l required to set up tool j on machine i.

5.5. Bound and Nature Variables.

$$\text{SA}_{il\,jt},\ S_{il\,jt} \in \{0, 1\}, \quad \forall i, l, j, t. \tag{31}$$

Constraint (31) indicates the binary nature of the setup S_{iljt} and setup amount SA_{iljt} variables. Finally, the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators is also subject to constraints (6), (8), (13a), (13b), (14)–(16), (18), and (19).

6. Case Study and Computational Experiments

The proposed base MILP model for lot sizing and scheduling on parallel flexible injection machines and the extended MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators was implemented in Python 3.8.2, using Pyomo [8] as an extensible python-based open-source optimization modeling language for linear programming. The performance of the proposed model was evaluated on a set of instances that reflect different characteristics of the real-world case of the automotive components industry under study. All the numerical tests were performed on a personal computer equipped with an Intel (R) Core (TM) I5-8500 @ 3.00 GHz Processor and 8 GB RAM. We used Python 3.8.2 and tested applying Gurobi 9.0, to solve the mixed integer linear programming model.

In the next section of data generation, it is described how the data are generated to run the computational experiments. The datasets generated to validate the proposed models correspond to small, medium, and large datasets. Sized datasets can be accessed through a link available at the end of the document. Finally, the last section presents the results of the computational experiments carried out.

6.1. Data Generation. For the data generation, we define various instance sets, including small, medium, and large data sizes. The small dataset corresponds to the minimum amount of data required to test the proposed model; the medium dataset allows to test the model with a reasonable number of parameters and variables to be solved by the model, approaching to the realistic view of the LSSP; and finally, the large dataset replicates the real amount of data managed by real world enterprises when solving the LSSP. For the computational experiments, all the datasets are built through considering the parameter values depicted on Table 4. The data values are created in the way that mostly represents real data from the automotive components industry; next, the data values are defined :

(i) The parameter a_j determines the total amount of tools j available for production; in this regard, only one unit of each tool is available, and this means that there are not duplicated tools to produce the same components.

(ii) The backorder cost (cb_k) and the coverage stockout cost (cst_k) are represented by a very high value (equal to M (99999)) in order to avoid customer missing parts in the model resolution.

(iii) Inventory costs (ci_k) are set with values uniformly distributed in given interval; $U(u_1, u_2)$ is a random variable which is uniformly distributed on $[u_1, u_2]$.

(iv) The stock coverage is defined in three days of demand. Nevertheless, in small datasets where the

Table 4: Generation of value for data parameters.

Parameter	Value
a_j	1
cb_k	99999
ci_k	U (0.1, 1)
cov_{kt}	1 when $T < 3$; Otherwise, 3 when $T > 3$
cr_{ij}	Random (5, 15)
cs_j	Random (45, 50)
cst_k	99999
d_{kt}	Random (15, 40) if T = first 5 periods of the week; Otherwise, 0 if $T = 6^{th}$ and $T = 7^{th}$ periods of the week
INV_{k0}	1
$INVMAX_k$	Random (10000, 20000)
$INVMIN_k$	1
nc_t	Random (I, $I+5$) 0 if $T = 7^{th}$ period of the week
np_{jk}	Random (2, 5)
p_{jk}	Random (2, 5)
tp_t	24 hours if T = first 5 periods of the week 16 hours if $T = 6^{th}$ period of the week 0 hours if $T = 7^{th}$ period of the week
r_{ij}	1
sla_{ilj}	1
scl_{ilj}	U (2.5, 3.5)
sls_l	I

number of periods is lower than three, we have considered one coverage day of demand ($cov_{kt} = 1$) in order not to have unfeasible solutions in the model resolution, in such a way that the model considers one period of future demand (d_{t+1}) to be produced during period t.

(v) Random [r_1, r_2] values denote a random integer value over the interval from r_1 to r_2. The following data parameters use Random [r_1, r_2] values: setup cost of a tool (cr_{ij}), setup cost of preparing a tool (cs_j), maximum inventory ($INVMAX_k$), and the amount of products no longer produced when a tool is set up (np_{jk}),

(vi) The amount of products produced when a tool is set up (p_{jk}) is also denoted as Random [r_1, r_2]. In this regard, $p_{jk} > 0$ indicates that product k is assigned to the tool j; otherwise, $p_{jk} = 0$. Parts are randomly assigned to tools, with the condition that each part must be assigned to one tool. Table 5 proposes an example of assignment on a small dataset composed of three tools and six products.

(vii) In order to generate the values for the demand, blocks of seven time periods corresponding to the 7 days of the week are considered. In this regard, the first fifth periods of the week will have demand values set as Random (15, 40); otherwise, $d_{kt} = 0$ on the sixth and seventh periods of the week, that is, on Saturday and Sunday.

(viii) The minimum inventory ($INVMIN_k$) is set as one unit for all the products k; accordingly, the initial inventory (INV_{k0}) is also set as one unit for all products k.

(ix) The amount of tool changes allowed (nc_t) is defined by Random (I, $I+5$), with I being the minimum number of machines changes allowed, that coincides with the total number of machines. Considering the same scheme of blocks of weeks divided in 7 periods, no tool changes are allowed in the 7^{th} period of the week, $nc_t = 0$; this is because on Sundays, the enterprise does not produce and uses it as a day of rest or for machine maintenance, etc.

(x) According to this last statement, the time available for production (tp_t) is 24 hours for the first 5^{th} periods of the week and 16 hours for the 6^{th} period of the week. No time production is available for the 7^{th} period of the week.

(xi) $r_{ij} = 1$ indicates that all the tools can be setup on all the machines.

The aforementioned parameters and values are defined for the base model. The three parameters added for the extended model, which considers set up common operators, are described below (see Table 4):

(i) $sla_{ilj} = 1$ indicates that one setup-type operator is required to setup the tool j on machine i

(ii) The cost of type operator l to setup the tool j on machine i (scl_{ilj}) is set with values uniformly distributed in the given interval; $U(u_1,u_2)$ is a random variable which is uniformly distributed on $[u_1,u_2]$

(iii) Finally, the number of available workers of type operator l available (sls_l) must be as much as the number of machines I

In addition, we attach the link to the synthetic data generator, so that the model can be reproduced in future research; https://bit.ly/3iwfObA.

6.2. Results and Computational Experiments. Using the synthetic data generator mentioned above, a set of experiments have been conducted to validate the two proposed models: (i) a base LSSP model, an MILP model for lot sizing and scheduling on parallel flexible injection machines and (ii) an extended LSSP model, an MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators.

In order to give the reader a clear insight of the input data parameter values and the output data results once implemented the proposed MILP, we include an example of a small dataset size. The input data for the base LSSP model are presented at https://bit.ly/3p3IFqo. The small dataset of the base model is characterized by having 2 machines, 4 tools, 6 parts, and 3 periods. The results obtained with the decision variables in the MILP model for lot sizing and scheduling on parallel flexible injection machines are presented in Tables 6 and 7.

According to the results obtained in the MILP model for lot sizing and scheduling on parallel flexible injection

Table 5: Amount of products produced when a tool is set up.

j	k	p_{jk}
1	4	4
2	1	2
2	5	2
2	6	2
3	2	2
4	3	3

Table 6: Small dataset results of the base LSSP MILP: lot sizing, inventories, coverage, stockout, and backorders.

k	t	X_{kt}	Xnp_{kt}	X'_{kt}	ST_{kt}	INV_{kt}	B_{kt}
1	1	48	0	48	2	21	0
1	2	27	0	27	0	25	0
1	3	0	0	0	0	1	0
2	1	0	0	0	33	1	35
2	2	48	0	48	30	1	21
2	3	48	0	48	0	1	4
3	1	72	0	72	0	39	0
3	2	0	0	0	11	19	0
3	3	0	0	0	0	1	12
4	1	0	0	0	14	1	30
4	2	0	0	0	32	1	45
4	3	96	0	96	0	19	0
5	1	48	0	48	8	31	0
5	2	41	0	41	0	33	0
5	3	0	0	0	0	1	0
6	1	48	0	48	19	15	0
6	2	48	0	48	5	29	0
6	3	0	0	0	0	1	6

Table 7: Small dataset results of the base LSSP MILP: scheduling.

i	j	t	SA_{ijt}	S_{ijt}	TP_{ijt}
1	2	1	1	1	24
1	2	2	0	1	24
1	1	3	1	1	24
2	4	1	1	1	24
2	3	2	1	1	24
2	3	3	0	1	24

machines, all the available capacity is occupied in the three defined periods. Notwithstanding, demand was higher than the available capacity, and therefore, the model had to delay the demand of products $k = 2, 3, 4$. Figure 3 shows the Gantt chart showing the schedule obtained after applying the base MILP model. Each row represents a machine i, and each rectangle a tool j; inside the rectangle, we have indicated the parts k and processing time t of each tool j.

The input data for the extended LSSP model are presented at https://bit.ly/3iA5hfq, and it is characterized by considering 2 machines, 4 tools, 6 parts, 3 periods, and 4 type operators. The results obtained with the decision variables in the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators are presented in Tables 8 and 9.

According to the obtained results, all the available capacity is occupied along the three defined periods. Nevertheless, demand was higher than the available capacity, and therefore, the model had to delay the demand of both product $k = 2, 3, 4, 6$. The solution provided in the scheduling shows that the two type operators l are assigned to change the tools j.

The experimental results obtained with the tests performed for all the datasets employed in the validation for both the base model (Table 10) and the extended model in which the labor index is added (Table 11) are provided below. Four instances are generated of each dataset size. The synthetic instances have been generated through the previously proposed generator: https://bit.ly/3iwfObA.

The computational results (Table 10) show that the MILP model for lot sizing and scheduling on parallel flexible injection machines in small (S1, S2, S3, and S4) and medium (M1, M2, M3, and M4) instances achieve optimal results (GAP = 0%) in, at maximum, 12.6 seconds. With regards to the large instances generated (L1, L2, L3, and L4), the calculation time on average is set as 3.3 hours finding a near optimal solution, with an average 0.52% GAP. In this regard, it is worth highlighting that the large dataset L1 achieves the optimal solution in a very reduced computational time of 44 seconds.

The computational results (Table 11) show that the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators in small (S1, S2, S3, and S4) and medium (M1, M2, M3, and M4)

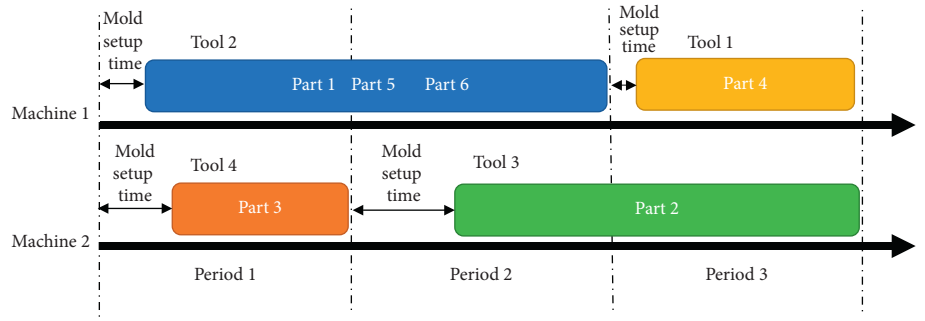


Figure 3: Gantt chart base MILP model for lot sizing and scheduling on parallel flexible injection machines.

Table 8: Small dataset results of the extended LSSP MILP, with setup common operators: lot sizing, inventories, coverage, stockout, and backorders.

k	t	X_k	Xnp_{kt}	X'_{kt}	ST_{kt}	INV_{kt}	B_{kt}
1	1	48	0	48	2	21	0
1	2	27	0	27	0	25	0
1	3	0	0	0	0	1	0
2	1	0	0	0	33	1	35
2	2	48	0	48	30	1	21
2	3	48	0	48	0	1	4
3	1	72	0	72	0	39	0
3	2	0	0	0	11	19	0
3	3	0	0	0	0	1	12
4	1	0	0	0	14	1	30
4	2	0	0	0	32	1	45
4	3	96	0	96	0	19	0
5	1	48	0	48	8	31	0
5	2	41	0	41	0	33	0
5	3	0	0	0	0	1	0
6	1	48	0	48	19	15	0
6	2	48	0	48	5	29	0
6	3	0	0	0	0	1	6

Table 9: Small dataset results of the base LSSP MILP: scheduling.

i	l	j	t	S_{iljt}	SA_{iljt}	TP_{ijt}
1	1	2	1	1	1	24
1	2	2	1	1	1	24
1	1	2	2	1	0	24
1	2	2	2	1	0	24
1	1	1	3	1	1	24
1	2	1	3	1	1	24
2	1	4	1	1	1	24
2	2	4	1	1	1	24
2	1	3	2	1	1	24
2	2	3	2	1	1	24
2	1	3	3	1	0	24
2	2	3	3	1	0	24

instances achieve optimal results (GAP = 0%) in, at maximum, 77.7 seconds. With regards to the large instances generated (L1, L2, L3, and L4), the calculation time on average is set as 2.8 hours finding a very near optimal solution, with an average 3% GAP. In this regard, it is worth highlighting that the large dataset L1 achieves the optimal solution in a very reduced computational time of 208.83 seconds.

The experiments carried out for both the base model and the extended model that consider common setup operators

Table 10: Experimental results for the MILP model for lot sizing and scheduling on parallel flexible injection machines.

Dataset	I	J	K	T	Number of constraints	Number of variables	Number of binary variables	Number of integer variables	Number of continuous variables	Number of nonzeros	Objective value	Termination condition	Time (sec)	GAP (%)
S1	2	4	6	3	253	180	48	48	84	548	30699918	Optimal	0.15767312	0.00
S2	4	6	8	3	529	360	144	144	72	1240	13400191	Optimal	0.13881159	0.00
S3	6	8	16	3	1037	720	288	288	144	2672	11100467	Optimal	0.13907146	0.00
S4	8	10	22	3	1617	1116	480	480	156	4372	8100579	Optimal	0.13229871	0.00
M1	10	12	24	14	10882	7056	3360	3360	336	29232	75400153.9	Optimal	12.6211112	0.00
M2	12	14	28	14	14630	9408	4704	4704	0	39872	162199368	Optimal	11.8187988	0.00
M3	14	16	32	14	18930	12096	6272	6272	−448	52160	129000642	Optimal	6.2376368	0.00
M4	16	18	36	14	23782	15120	8064	8064	−1008	66096	109300097	Optimal	12.3485208	0.00
L1	18	20	40	14	29186	18480	10080	10080	−1680	81680	226999024	Optimal	44.0416777	0.00
L2	20	40	60	14	61754	38640	22400	22400	−6160	164880	698898867	maxTimeLimit	10801.9226	0.53
L3	25	45	70	14	85269	53130	31500	31500	−9870	230910	889697348	maxTimeLimit	18000.7021	1.23
L4	30	50	80	21	169221	104580	63000	63000	−21420	462680	764199835	maxTimeLimit	18025.0699	0.32

Table 11: Experimental results for the MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators.

Dataset	I	J	K	L	T	Number of constraints	Number of variables	Number of binary variables	Number of integer variables	Number of continuous variables	Number of nonzeros	Objective value	Termination condition	Time (sec)	GAP (%)
S1	2	4	6	2	3	392	228	96	96	36	880	30700163	Optimal	0.09	0.00
S2	4	6	8	2	3	904	504	288	288	−72	2224	9700582	Optimal	1.41	0.00
S3	6	8	16	2	3	1760	1008	576	576	−144	4736	23500652	Optimal	0.16	0.00
S4	8	10	22	2	3	2800	1596	960	960	−324	7860	18901051	Optimal	0.16	0.00
M1	10	12	24	4	14	36744	17136	13440	13440	−9744	100032	67803541	Optimal	19.20	0.00
M2	12	14	28	4	14	50596	23520	18816	18816	−14112	138992	99203043	Optimal	13.08	0.00
M3	14	16	32	4	14	66656	30912	25088	25088	−19264	184320	143003146	Optimal	39.02	0.00
M4	16	18	36	4	14	84924	39312	32256	32256	−25200	236016	146304008	Optimal	77.73	0.00
L1	18	20	40	6	14	156200	68880	60480	60480	−52080	432320	211505838	Optimal	208.83	0.00
L2	20	40	60	8	14	454244	195440	179200	179200	−162960	1212080	878439687	maxTimeLimit	10816.94	3.05
L3	25	45	70	12	14	950246	399630	378000	378000	−356370	2545660	912365839.8	maxTimeLimit	18032.57	4.47
L4	30	60	120	15	14	1893720	791280	756000	756000	−720720	5256960	1367518374	maxTimeLimit	10901.30	5.11

are valid for their application on solving LSSP models with real amount of data managed by real-world enterprises.

7. Discussion and Conclusions

This article addresses the LSSP applied to an automotive plastic components' enterprise. An MILP base model is proposed to deal with the lot-sizing and scheduling problem on parallel flexible injection machines to mainly minimize the setup, inventory, stockout, and backorder costs by taking into account injection molds as the main index to schedule parallel flexible injection machines. The MILP base model is extended to provide the enterprise under study with a more realistic solution that considers setup common operators. Therefore, the extended LSSP model, an MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators, is presented. This produces a model that adapts to the restrictions of the company under study, an automotive plastic components enterprise. The peculiarity of this model is that it takes injection molds as the main index to schedule parallel flexible injection machines by considering setup common operators. The novelty of both proposed models lies in our study considering many periods when modeling and running experiments to solve the LSSP. Moreover, the proposed MILP bears in mind stock coverage constraints, which are typical in the studied automotive supply chain industry context, and contemplates an objective function that allows idle times among molds, which is a fundamental characteristic for real cases and has been ignored by former studies.

Finally, this article validates the proposed MILP by performing experiments with different sized instances, including small, medium, and large datasets. The large dataset is characterized by replicating the amount of data used in the real enterprise that is the object of this study. The goodness of the model is evaluated with the computational time and the deviation of the obtained results as regards to the optimal solution.

This study is not without its limitations. The small and medium datasets are solved in both cases in very efficient computing times. The application of the proposed model using the large dataset is more limited in computational efficiency terms. To solve this problem, the literature indicates the generation of heuristics, metaheuristics, and matheuristics. This last type falls within the authors' future research options as far as the contemplated model herein is concerned. Thus, the first research line intends to improve the computational efficiency to solve the model by applying matheuristics, which would consist in solving the binary variables in both the base model base (S_{ijt} and SA_{ijt}) and its extended version (S_{iljt} and SA_{iljt}) by a metaheuristic technique, e.g., genetic algorithms, taboo search, and simulated annealing. The metaheuristics result will be provided as input data for the MILP. The following research lines focus on improving the model by considering new constraints that will be very useful for the real firm. In this way, the second future research line is a second extension of the base model that bears in mind the availability of materials. For this purpose, inventory equations are to be added that consider the list of materials needed to manufacture end products. Finally, for the third future research line, a third extension of the model is proposed by considering space limitations in a warehouse's volume. In this way, part volumes are calculated to meet both stock coverage and limited warehouse space, contemplated from the premise that all parts have different volume requirements. This means that the model and its third extension (or third-generation) will be capable of meeting the stock coverage of large-sized parts and stocking small-sized ones so that when they have to be produced, the production resources related to bigger pieces will be used. The third-generation model will allow bigger-sized parts to be stored, which will be left at the stock coverage level by calculating the quantity of smaller parts whose coverage can be extended.

Data Availability

The instances generated and analyzed during the study are available at (1) the base LSSP model, an MILP model for lot sizing and scheduling on parallel flexible injection machines: https://bit.ly/3p3IFqo. (2) The extended LSSP model, an MILP model for lot sizing and scheduling on parallel flexible injection machines with setup common operators: https://bit.ly/3iA5hfq. The algorithm developed for generating the synthetic datasets is available at https://bit.ly/3iwfObA.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the Conselleria de Educación, Investigación, Cultura y Deporte-Generalitat Valenciana for hiring predoctoral research staff with Grant no. ACIF/2018/170 and European Social Fund with Grant Operational Program of FSE 2014-2020, the Valencian Community, and the authors would like to acknowledge the support of the researchers participating in the collaborative projects 'Cloud Collaborative Manufacturing Networks' (C2NET) (http://c2net-project.eu/), which has received funding from the EU Horizon 2020 Research and Innovation Programme with grant agreement no. 63690, and "Zero Defects Manufacturing Platform" (ZDMP) (http://www.zdmp.eu), which has received funding from the EU Horizon 2020 Research and Innovation Programme with grant agreement no. 825631.

References

[1] R. Gujjula, S. Werk, and H.-O. Günther, "A heuristic based on Vogel's approximation method for sequencing mixed-model assembly lines," *International Journal of Production Research*, vol. 49, no. 21, pp. 6451–6468, 2011.

[2] S.-i. Kim, J. Han, Y. Lee, and E. Park, "Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum," *Computers & Operations Research*, vol. 37, no. 2, pp. 302–314, 2010.

[3] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.
[4] S. Echchakoui and N. Barka, "Industry 4.0 and its impact in plastics industry: a literature review," *Journal of Industrial Information Integration*, July, vol. 20, , p. 100172, 2020.
[5] H2020 Project C2NET, "Cloud Collaborative Manufacturing Networks' (C2NET)," 2017.
[6] R. Sanchís, B. Andrés, R. Poler, J. Mula, and M. Díaz-Madroñero, "La solución de optimización C2NET," *Dirección Y Organización*, vol. 64, no. 64, pp. 36–41, 2018.
[7] K. P. Abdul Nazar and V. Madhusudanan Pillai, "Mixed-model sequencing problem under capacity and machine idle time constraints in JIT production systems," *Computers & Industrial Engineering*, vol. 118, pp. 226–236, 2018.
[8] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo-Optimization Modeling in Python*, Springer Publishing Company, Berlin, Germany, 1st edition, 2012.
[9] Y. Á. Ríos-Solís, O. J. Ibarra-Rojas, M. Cabo, and E. Possani, "A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production," *European Journal of Operational Research*, vol. 284, no. 3, pp. 861–873, 2020.
[10] O. J. Ibarra-Rojas, R. Z. Ríos-Mercado, Y. A. Rios-Solis, and M. A. Saucedo-Espinosa, "A decomposition approach for the piece-mold-machine manufacturing problem," *International Journal of Production Economics*, vol. 134, no. 1, pp. 255–261, 2011.
[11] Y. Li, Z. Yang, G. Li, D. Zhao, and W. Tian, "Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1565–1575, Feb. 2019.
[12] L. Guimarães, D. Klabjan, and B. Almada-Lobo, "Modeling lotsizing and scheduling problems with sequence dependent setups," *European Journal of Operational Research*, vol. 239, no. 3, pp. 644–662, 2014.
[13] H. Stadtler, "Multi-level single machine lot-sizing and scheduling with zero lead times," *European Journal of Operational Research*, vol. 209, no. 3, pp. 241–252, 2011.
[14] H. Meyr and M. Mann, "A decomposition approach for the general lotsizing and scheduling problem for parallel production lines," *European Journal of Operational Research*, vol. 229, no. 3, pp. 718–731, 2013.
[15] C. Wolosewicz, S. Dauzère-Pérès, and R. Aggoune, "A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem," *European Journal of Operational Research*, vol. 244, no. 1, pp. 3–12, 2015.
[16] R. J. W. James and B. Almada-Lobo, "Single and parallel machine capacitated lotsizing and scheduling: new iterative MIP-based neighborhood search heuristics," *Computers & Operations Research*, vol. 38, no. 12, pp. 1816–1825, 2011.
[17] M. Díaz-Madroñero, J. Mula, B. Andres, R. Poler, and R. Sanchis, "A capacitated lot-sizing and scheduling model for the bi-part injection moulding problem," 2017.
[18] J. Mula, A. C. Lyons, J. E. Hernández, and R. Poler, "An integer linear programming model to support customer-driven material planning in synchronised, multi-tier supply chains," *International Journal of Production Research*, vol. 52, no. 14, pp. 4267–4278, 2014.

Hindawi
Complexity
Volume 2021, Article ID 8822765, 9 pages
https://doi.org/10.1155/2021/8822765

Research Article

Sizing a Hybrid Renewable Energy System by a Coevolutionary Multiobjective Optimization Algorithm

Wenhua Li,[1] **Guo Zhang**,[1] **Xu Yang**,[1] **Zhang Tao**,[1,2] **and Hu Xu**[3]

[1]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*Hunan Key Laboratory of Multi-Energy System Intelligent Interconnection Technology, Changsha 410073, China*
[3]*State Key Laboratory of Astronautic Dynamics, Xi'an 710043, China*

Correspondence should be addressed to Wenhua Li; liwenhua1030@aliyun.com

Received 7 September 2020; Revised 15 September 2020; Accepted 23 January 2021; Published 10 February 2021

Academic Editor: Christos Volos

Hybrid renewable energy system (HRES) arises regularly in real life. By optimizing the capacity and running status of the microgrid (MG), HRES can decrease the running cost and improve the efficiency. Such an optimization problem is generally a constrained mixed-integer programming problem, which is usually solved by linear programming method. However, as more and more devices are added into MG, the mathematical model of HRES refers to nonlinear, in which the traditional method is incapable to solve. To address this issue, we first proposed the mathematical model of an HRES. Then, a coevolutionary multiobjective optimization algorithm, termed CMOEA-c, is proposed to handle the nonlinear part and the constraints. By considering the constraints and the objective values simultaneously, CMOEA-c can easily jump out of the local optimal solution and obtain satisfactory results. Experimental results show that, compared to other state-of-the-art methods, the proposed algorithm is competitive in solving HRES problems.

1. Introduction

Energy shortages and environmental pollution problems have become increasingly severe in recent years. Microgrid (MG) technology [1] has received more and more attention to improving the rate of renewable energy usage, especially in large-scale wind and solar energy. MG is a small power supply and consumption system that contains a variety of components, e.g., electrical loads, energy storage devices, and renewable energy. So far, there have been many pieces of research on microgrid energy management. However, how to plan the capacity of the microgrid and consider the running operation of the microgrid in the design stage has become an urgent problem.

Hybrid renewable energy system (HRES) [2] aims to optimize the installed capacity of components and energy management of the MG system during the planning phase. The main problem is to determine a set of optimal configurations, including whether to install a component and the capacity to install [3]. The main problem in microgrid energy management is how to formulate a power generation and scheduling plan without affecting the stable operation of the system.

To address the above problem, many studies have proposed and improved the HRES model [4, 5]. Generally speaking, this problem is mixed-integer programming, so it can be optimized by traditional solving tools, e.g., branch-and-cut [6]. However, with the update of microgrid components, the HRES model has increasingly shown nonlinear characteristics, which cannot be solved directly by traditional linear programming methods. It is reported that through the transformation, which transfers the nonlinear constraints and objective functions to their linear form, some simple nonlinear problems can be solved by traditional linear programming methods. However, such transformation is limited and loses the information of the original problems. On the other hand, the HRES problem is a typical multiobjective optimization problem [3]. Decision makers (DMs) need to consider the full life cycle cost, power reliability, pollution emission, and so on simultaneously [7].

The traditional linear programming method usually converts multiobjective optimization problems to single-objective one by summing up objectives with a group of given weights [8]. That is, DM gives a set of weight vectors in advance, and then a scalar function is constructed for optimization. Although this method is simple, it has many drawbacks. Firstly, due to the inconsistency of various objective functions, it is difficult to determine the weights, resulting in poor robustness of the single-objective optimization problem. The use of larger weights will amplify the influence of the objective function noise, leading to unilateral preference. Secondly, it is often difficult for decision makers to give appropriate preference weights in advance [9].

Evolutionary multiobjective optimization (EMO) algorithm has gained more and more attention in dealing with real-world engineering problems with nonlinear objective functions and constraints [10]. Multiobjective optimization problems (MOPs) which require the simultaneous optimization of multiple objectives are common in real-world applications. Without loss of generality, a MOP can be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\}, \\ \text{s.t.} \quad & \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega, \end{aligned} \tag{1}$$

where Ω denotes the feasible decision space, m is the number of objectives, and $\mathbf{x}$ is a decision vector consisting of n decision variables x_i. A solution $\mathbf{x_a}$ is said to Pareto dominate another solution $\mathbf{x_b}$ iff $\forall i = 1, 2, \ldots, m$, $f_i(\mathbf{x_a}) \leq f_i(\mathbf{x_b})$ and $\exists j = 1, 2, \ldots, m$, $f_j(\mathbf{x_a}) < f_j(\mathbf{x_b})$. The images of Pareto optimal solution set (PS) in the objective space are termed the Pareto optimal front (PF).

So far, a large number of works have been done to find a well-distributed PF of a MOP without constraints, which can be roughly categorized as (i) Pareto dominance-based algorithms, e.g., NSGA-II [11] and SPEA2 [12], (ii) indicator-based algorithms, e.g., IBEA [13], and (iii) decomposition-based methods, e.g., MOEA/D [14, 15]. Although the proposed EMO algorithms can well solve the benchmark problems without constraints [16], they are not specially designed for MOPs with constraints. To deal with constraints by EMO algorithms, many approaches have been proposed. Among them, penalty function [17, 18] may be the most representative and easiest way, which introduces a penalty term into the objective function to penalize constraint violations on a minimization problem. The introduction of the penalty term enables us to transform a constrained optimization problem into an unconstrained one. However, it is very difficult to strike the right balance between objective and penalty functions. Other approaches like ϵ-constraints [19] and stochastic ranking [20] can somehow deal with the constraints. Another effective way to deal with the constraints is to modify the dominance relationship [21], which has been proved effective and efficient. Specifically, there are two basic rules to calculate the dominance relationship. Solutions satisfying the constraints dominate unsatisfied solutions. If both solutions satisfy the constraints, then use the basic rule to compare their objective values.

In addition, it is still an open issue to solve mixed-integer programming problem through EMO [22, 23]. Compared to continuous problems, the decision spaces of mixed-integer problems are separated. As shown in Figure 1, a very simple example is given with only two integer variables. In this case, each feasible region contains only one solution. The number of feasible regions will exponentially increase as the number of decision variables grows. In general, we set the population size to 100 to optimize, which is largely insufficient for the whole decision space. Moreover, since feasible regions in the decision space are separated, it is difficult for EMO algorithms to continuously search the whole decision space.

In this paper, we first proposed the mathematical model of an HRES. We consider the building and running cost of this system as well as the energy supply stability. This HRES model is a nonlinear and mixed-integer multiobjective optimization problem. To solve this problem, we proposed a coevolutionary EMO algorithm with modified dominance relationship to deal with the constraints, termed CMOEA-c. Specifically, there is an assistant archive which aims to explore the whole decision space and help jump out of the local optimal solution. In addition, we compared the performance of the proposed algorithm with other state-of-the-art algorithms. Experimental results show that our model and algorithm are effective and efficient.

The rest of this paper is structured as follows. A mathematical model of a hybrid renewable energy system is proposed and specifically explained in Section 2. In Section 3, the proposed coevolutionary multiobjective optimization algorithm is illustrated, followed by Section 4, which shows the experiment setting and the result analysis. Finally, a conclusion of this work is given in Section 5.

2. Mathematical Model of the Hybrid Renewable Energy System (HRES)

A typical HRES model includes the following components, i.e., the external electricity grid, renewable power generation, user load, diesel generator (DE), and energy storage system (ESS), as shown in Figure 2. We need to first determine the installed capacities of these components. Then, we can optimize the control strategy of ESS and DE to minimize the running cost of this HRES.

2.1. Constraints. Due to the randomness and intermittent nature of renewable energy, the power supply is often unstable [24], and it is difficult to meet energy demand all the time with renewable energy alone. With the help of ESS, the gap between supply and demand can be eased to a certain extent. In order to improve the reliability of power supply, MG must be connected to an external power grid. In this process, to ensure safety, the maximum transmission capacity is limited within a certain range.

$$X_{\text{sell}} P_{\text{grid}}^{\min} \leq P_{\text{grid}}(t) \leq X_{\text{sell}} P_{\text{grid}}^{\max}, \tag{2}$$

where $P_{\text{grid}}(t)$ is the exchanged power between external grid and MG. $P_{\text{grid}}^{\min}$ and $P_{\text{grid}}^{\max}$ represent the minimum and maximum power of the interaction respectively. X_{sell} is a

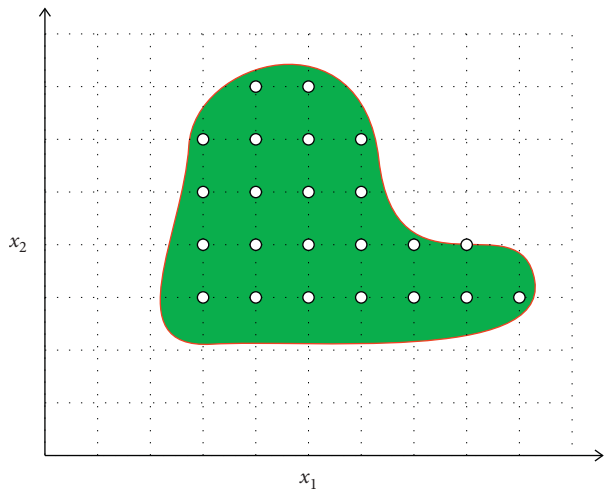


Figure 1: Decision space of a mixed-integer programming problem.

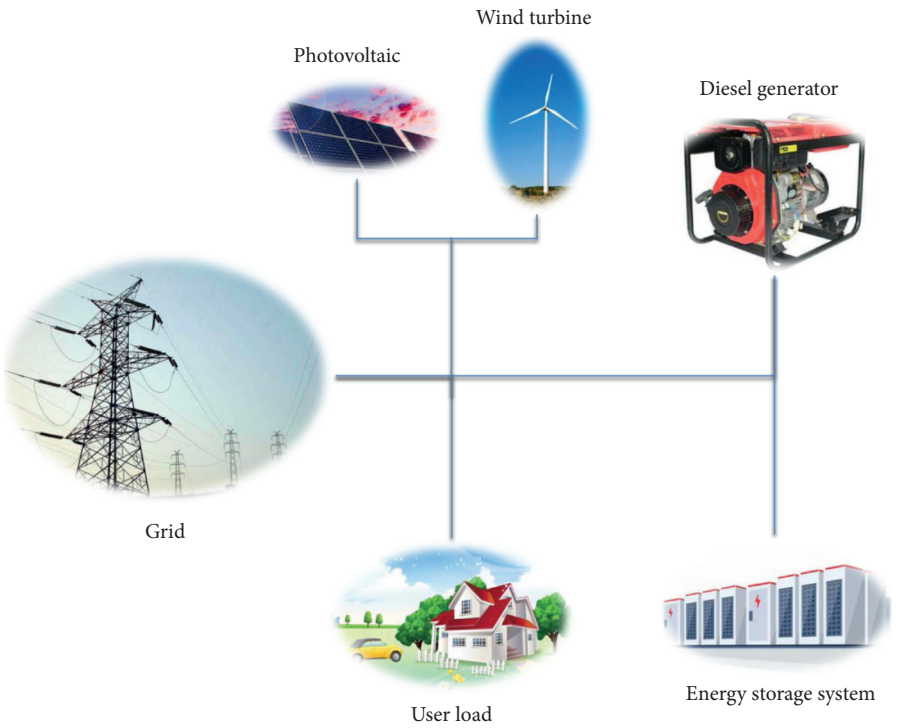


Figure 2: Physical structure of an HRES.

binary variable that indicates whether the MG can sell electricity to the grid. $P_{\text{grid}}^{\min}$ could be negative which means that the MG can sell electricity to the external grid.

Renewable energy includes photovoltaic and wind turbine; the main constraint is the minimum and maximum installed capacity.

$$\begin{aligned} A_{\min}^{\text{pv}} \cdot X_{\text{pv}} \le A_{\text{pv}} \le A_{\max}^{\text{pv}} \cdot X_{\text{pv}}, \\ P_{\text{pv}}(t) \le A_{\text{pv}} C p_{\text{rat}}^{\text{pv}}, \end{aligned} \tag{3}$$

where A_{pv} is the installed area of photovoltaic; X_{pv} is a binary variable presenting if the photovoltaic equipment is installed or not; and $Cp_{\text{rat}}^{\text{pv}}$ is the transfer efficiency of photovoltaic. The constraints of the wind turbine can be expressed as

$$\begin{aligned} N_{\min}^{\text{wt}} \cdot X_{\text{wt}} \le N_{\text{wt}} \le N_{\max}^{\text{wt}} \cdot X_{\text{wt}}, \\ P_{\text{wt}}(t) \le C p_{\text{rat}}^{\text{wt}} \cdot N_{\text{wt}}, \end{aligned} \tag{4}$$

where N_{wt} and $Cp_{\text{rat}}^{\text{wt}}$ represent the installed number and rated capacity of the wind turbine, respectively. X_{wt} is the selecting state of the wind turbine, where $X_{\text{wt}} = 0$ means that wind turbine is not installed in this HRES.

Similarly, the constraints of diesel generator are listed as follows:

$$N_{\min}^{\mathrm{de}} \cdot X_{\mathrm{de}} \le N_{\mathrm{de}} \le N_{\max}^{\mathrm{de}} \cdot X_{\mathrm{de}},$$
$$P_{\mathrm{de}}(t) \le Cp_{\mathrm{rat}}^{\mathrm{de}} \cdot N_{\mathrm{de}}. \tag{5}$$

Energy storage system (ESS) in an MG can not only store excess power but also supply power as a backup when the power generation is insufficient. With the help of the ESS, HRES can run more flexibly and stably. The main constraints are shown as follows:

$$\begin{aligned} P_{\mathrm{ess}}^{\min} \cdot X_{\mathrm{ess}} &\le P_{\mathrm{ess}}(t) \le P_{\mathrm{ess}}^{\max} \cdot X_{\mathrm{ess}}, \\ P_{\mathrm{ess}}^{\max} &= Cp_{\mathrm{rat}}^{\mathrm{ess}} \cdot N_{\mathrm{ess}}, \\ E_{\mathrm{ess}}^{\min} \cdot X_{\mathrm{ess}} &\le E_{\mathrm{ess}}(t) \le E_{\mathrm{ess}}^{\max} \cdot X_{\mathrm{ess}}, \\ E_{\mathrm{ess}}(t+1) &= E_{\mathrm{ess}}(t)\left(1 - \epsilon_{\mathrm{ess}}\right) + \eta_{\mathrm{ess}} P_{\mathrm{ess}}(t) \Delta t, \end{aligned} \tag{6}$$

where $P_{\mathrm{ess}}^{\min}$ and $P_{\mathrm{ess}}^{\max}$ are the minimum and maximum power of charging and discharging, respectively. $E_{\mathrm{ess}}(t)$ means the energy storage capacity at t. In addition, ϵ_{ess} is the self-discharge rate of ESS, which is provided by the manufacturer.

Parameter η_{bess} denotes the charging and discharging efficiency of the ESS, which changes according to the state (charging and discharging) of the ESS.

$$\eta_{\mathrm{bess}} = \begin{cases} \eta_{\mathrm{bess}}^{c}, & P_{\mathrm{bess}}(t) > 0, \\ \eta_{\mathrm{bess}}^{d}, & P_{\mathrm{bess}}(t) \le 0, \end{cases} \tag{7}$$

where η_{bess}^{c} and η_{bess}^{d} denote the charge/discharge efficiency of the energy storage system ranging from 0 to 1.

The power balance constraint refers to balancing the power supply and demand as shown in the following equation:

$$P_{\mathrm{pv}}(t) + P_{\mathrm{wt}}(t) + P_{\mathrm{de}}(t) + P_{\mathrm{grid}}(t) + P_{\mathrm{lack}}(t) = P_{\mathrm{load}}(t) + P_{\mathrm{ess}}(t), \tag{8}$$

where $P_{\mathrm{pv}}(t)$, $P_{\mathrm{wt}}(t)$, $P_{\mathrm{de}}(t)$, $P_{\mathrm{grid}}(t)$, $P_{\mathrm{load}}(t)$, and $P_{\mathrm{ess}}(t)$ represent the power of photovoltaic, wind turbine, diesel generator, interaction with external grid, users' load, and the energy storage system, respectively.

2.2. Objective Functions. In this HRES model, we consider two objective functions, one is to minimize the life cycle cost, and the other is to minimize the rate of lack of electricity. Specifically, the life cycle cost includes the investment cost and maintenance cost, which is calculated as

$$\begin{aligned} C_{\mathrm{INV}} &= C_{\mathrm{INV}}^{\mathrm{pv}} + C_{\mathrm{INV}}^{\mathrm{wt}} + C_{\mathrm{INV}}^{\mathrm{ess}} + C_{\mathrm{INV}}^{\mathrm{de}}, \\ C_{\mathrm{INV}}^{\mathrm{pv}} &= \mathrm{CRF}^{\mathrm{pv}} \cdot Cp_{\mathrm{rat}}^{\mathrm{pv}} \cdot A_{\mathrm{pv}} \cdot C_{\mathrm{pv}}, \\ C_{\mathrm{INV}}^{\mathrm{wt}} &= \mathrm{CRF}^{\mathrm{wt}} \cdot Cp_{\mathrm{rat}}^{\mathrm{wt}} \cdot N_{\mathrm{wt}} \cdot C_{\mathrm{wt}}, \\ C_{\mathrm{INV}}^{\mathrm{de}} &= \mathrm{CRF}^{\mathrm{de}} \cdot Cp_{\mathrm{rat}}^{\mathrm{de}} \cdot N_{\mathrm{de}} \cdot C_{\mathrm{de}}, \\ C_{\mathrm{INV}}^{\mathrm{ess}} &= \mathrm{CRF}^{\mathrm{ess}} \cdot Cp_{\mathrm{rat}}^{\mathrm{ess}} \cdot N_{\mathrm{ess}} \cdot C_{\mathrm{ess}}, \end{aligned} \tag{9}$$

where C_{INV} and $C_{\mathrm{equipment}}$ are the total and unit installed costs of each component. In this study, we only consider two-day energy management. Thus, CRF is introduced to calculate the annual cost, which is given by decision makers.

The maintenance cost of HRES can be expressed as

$$\begin{aligned} C_{\mathrm{MTN}} &= C_{\mathrm{MTN}}^{\mathrm{pv}} + C_{\mathrm{MTN}}^{\mathrm{wt}} + C_{\mathrm{MTN}}^{\mathrm{ess}} + C_{\mathrm{MTN}}^{\mathrm{de}} + C_{\mathrm{op}}^{\mathrm{grid}}, \\ C_{\mathrm{MTN}}^{\mathrm{pv}} &= \mathrm{CRF}^{\mathrm{pv}} \cdot P_{\mathrm{pv}} \cdot C_{\mathrm{mtn}}^{\mathrm{pv}}, \\ C_{\mathrm{MTN}}^{\mathrm{wt}} &= \mathrm{CRF}^{\mathrm{wt}} \cdot P_{\mathrm{wt}} \cdot C_{\mathrm{mtn}}^{\mathrm{wt}}, \\ C_{\mathrm{MTN}}^{\mathrm{de}} &= \mathrm{CRF}^{\mathrm{de}} \cdot P_{\mathrm{de}} \cdot C_{\mathrm{mtn}}^{\mathrm{de}}, \\ C_{\mathrm{MTN}}^{\mathrm{ess}} &= \mathrm{CRF}^{\mathrm{ess}} \cdot P_{\mathrm{ess}} \cdot C_{\mathrm{mtn}}^{\mathrm{ess}}, \\ C_{\mathrm{op}}^{\mathrm{Grid}} &= C_{\mathrm{Pur}}^{\mathrm{grid}} \cdot P_{\mathrm{Pur}}^{\mathrm{grid}} - C_{\mathrm{Sal}}^{\mathrm{grid}} \cdot P_{\mathrm{Sal}}^{\mathrm{grid}}, \end{aligned} \tag{10}$$

where $C_{\mathrm{op}}^{\mathrm{Grid}}$ is the cost of purchasing and selling electricity with the grid.

Another objective is to minimize the rate of lack of electricity. Therefore, the objective functions of HRES are written as follows:

$$\begin{aligned} \mathrm{Min}\, f_1 &= C_{\mathrm{INV}} + C_{\mathrm{MTN}}, \\ \mathrm{Min}\, f_2 &= \sum P_{\mathrm{lack}}(t). \end{aligned} \tag{11}$$

Generally speaking, these two objectives are conflicting, which means we cannot simply optimize one objective. Reasonably, the higher the cost of investment and maintenance, the lower the lack of electricity.

3. The Constrained Coevolutionary Multiobjective Optimization Algorithm

3.1. Motivation and Framework. There are many approaches to deal with constraints in a MOP as we have reviewed in Section 1. These approaches can obtain favorable results on benchmark problems. However, real-world engineering problems are generally complex and multimodal. Generally, the feasible regions of these problems are usually separated from each other in objective space and decision space. Figure 3 explains the objective space of a two-objective real-world engineering problem, where the red line means the Pareto front. As shown in Figure 3, there are two feasible regions B and D in the objective space, separated by infeasible region C. For traditional EMO algorithms with constraint-handling strategies, it is really hard for populations to enter into region B due to the existence of infeasible region C. Generally speaking, at the beginning of the algorithm run, most of the solutions will randomly locate outside region C. As the algorithms adopt the constraint-handling methods, solutions will converge to the bound of region C and D and keep exploring region D. As a result, the algorithm is stuck into the local optimal solution and it is extremely hard to jump out.

To address this issue, we proposed a constrained coevolutionary multiobjective optimization algorithm, termed CMOEA-c, which introduces an archive to help explore the whole decision space. The thoughts of coevolution can be widely found in some state-of-the-art EMO algorithms [25–27], which is proved effective and efficient. Specifically, we do not process the constraints in this archive. This archive aims to find the Pareto optimal front of a normal MOP without considering the constraints. Therefore, solutions in the archive

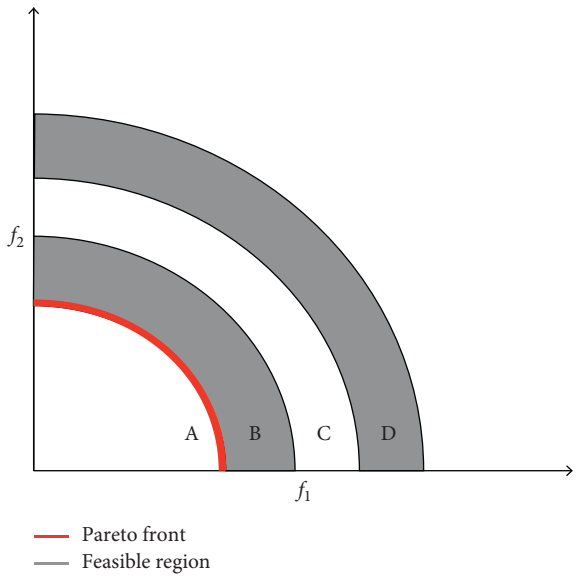


Figure 3: Illustration of a constraint multiobjective problem, where the feasible regions are separate.

will quickly converge to the true Pareto front, namely, the red line in Figure 3. In the beginning of the evolution process, solutions in the main population and the archive are randomly generated and may locate on regions C and D. As the algorithm runs, solutions in main population will converge to the optimal solution of region D, which is similar to that of traditional constrained EMO. As a comparison, since solutions in the archive do not consider the constraints, they will directly go through region C. Once the solutions in the archive enter into region B, this information will pass to the main population through the sharing function. As a result, the main population will quickly move to region B and then converge to the true Pareto optimal front. The framework of the proposed algorithm is explained in Algorithm 1.

As we can see from Algorithm 1, the main population and the introduced archive will coevolve by sharing their offspring (line 8 – 9). Therefore, solutions in region B can be added to the main population and help to converge to the true Pareto front. We will explain the selection strategy in the following section in detail.

3.2. Environmental Selection. After combining with the offspring, we need to update the population and the assistant archive by environmental selection, namely, maintain the population size to a certain number, shown in Algorithm 2. There is one major difference between the update process of the population and the archive. Specifically, there are two steps. Firstly, we need to calculate the dominance ranking information and crowding distance of the population. Secondly, if the number of nondominated solutions exceeds N, then we truncate it to N according to crowding distance; otherwise, we will firstly choose all nondominated solutions and then select other solutions by crowding distance to make sure the population size is equal to N.

Notably, for the main population, we use modified dominance relationship. That is, a feasible solution dominates an infeasible solution; if two solutions are feasible, then we use objective values to calculate the dominance relationship. As for the assistant archive, we do not consider whether the solution is feasible or not. The calculation of the dominance relationship is the same as the traditional ones.

3.3. Solving HRES by CMOEA-c. The main problem to solve HRES problem by EMO algorithms is to encode a solution and design the mutation strategies. Since the HRES model in the work is complex, we use a structure to represent the solution, which can be seen in Figure 4.

As we can see from Figure 4, there are five parts describing a solution for an HRES. The first two parts indicate the installed information of an HRES, including whether to install the equipment or not and the installed capacity. Specifically, 0 means that the HRES does not select to install this equipment, and vice versa. Besides, the running state of this HRES is determined by the first four parts, where ESS running state and DE running state participate. Notably, some instrumental variables can help to calculate the objective function values but do not take part in the mutation, which are stored in the final part of the solution, e.g., wind power output, photovoltaic power output, and electricity prices. Specifically, the values of these variables change when the equipment select state and installed capacity change.

As we can see from Figure 4, there are several variable types in a solution for an HRES. Thus, simply applying the same mutation strategy to evolve the solution is low-efficient and unreasonable. To accelerate the searching process, we divide them into several parts and perform different mutation strategies. Specifically, for the equipment select states, we use a small mutation probability $p_m = 0.1$ because the change of equipment select state will significantly affect the running state of an HRES. In addition, for mixed-integer variables, we will perform integrity checking after the mutation to ensure the solution is feasible. The mutation probabilities for mixed-integer variables and continuous variables are 0.5 and 1, respectively.

4. Experiments

4.1. Parameter Setting. To examine the performance of CMOEA-c, we choose SRA [20] and SPEA2+$_{\text{pf}}$ (SPEA2+ [28] with penalty function) as the competitor algorithms. Specifically, SRA is the most representative algorithm using stochastic ranking to deal with the constraints; SPEA2+ is chosen as the latest MOEA where the penalty function is added to each objective directly. All parameters are set to default values according to the original paper for fairness. For all algorithms, population size is 100 and the number of function evaluation is 20000. Each experiment is run for 31 times and the mean value is adopted for comparison and presentation. The parameters of this HRES are listed in Table 1. Besides, the generation power of the wind turbine and photovoltaic is related to local weather, which is shown in Figure 5. Moreover, the purchasing price and selling price of electricity are given as well.

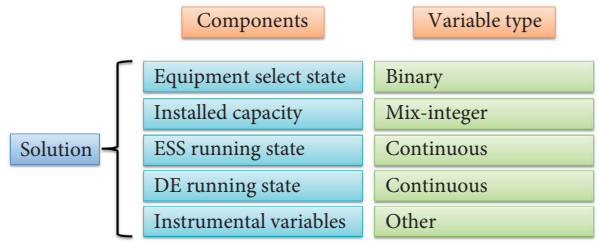


Figure 4: Illustration of solution encoding method and the variable type.

Input: Maximum generations MaxGen, population size N
Output: Nondominated set Pop
1 Pop $\longleftarrow$ Initialization(N)
2 Arc $\longleftarrow$ Initialization(N)
3 **While** gen $\leq$ MaxGen
4 $\quad$ $\text{MatingPool}_1 \longleftarrow$ TournamentSelection(Pop)
5 $\quad$ $\text{MatingPool}_2 \longleftarrow$ TournamentSelection(Arc)
6 $\quad$ $\text{OS}_1 \longleftarrow$ Variation(MatingPool_1)
7 $\quad$ $\text{OS}_2 \longleftarrow$ Variation(MatingPool_2)
8 $\quad$ Pop $\longleftarrow$ EnvSelection(Pop, OS_1, OS_2)
9 $\quad$ Arc $\longleftarrow$ EnvSelection(Arc, OS_1, OS_2)
10 **End While**

Algorithm 1: General framework.

Input: Joint population JointPop, population size N
Output: Updated population Pop
1 Ranking $\longleftarrow$ NondominatedSort(JointPop)
2 Crowd D is $\longleftarrow$ CalCrowd D is(JointPop)
3 NumNondominated$\leftarrow$number(Ranking == 1)
4 **If** NumNondominated $> N$ **then**
5 $\quad$ Pop $\longleftarrow$ JointPop(Ranking == 1)
6 $\quad$ Pop $\longleftarrow$ Pop(Crowd D is(1: N))
7 **else**
8 $\quad$ Pop $\longleftarrow$ JointPop(Ranking == 1)
9 $\quad$ Pop $\longleftarrow$ Pop $\cup$ Pop(Crowd D is(1: N − NumNondominated))
10 **End If**

Algorithm 2: Environmental selection.

4.2. Performance of the Proposed Algorithm. To examine the performance of CMOEA-c, we choose two constraint EMO algorithms as the competitors. It is worth mentioning that all experiments are executed 31 times independently. We choose the mean result among 31 times to present, which is shown in Figure 6.

As we can see from Figure 6, the Pareto front obtained by CMOEA-c is better than that of SRA. Moreover, the result obtained by SPEA2+$_{\text{pf}}$ is far away from the true Pareto front. This is because directly adding penalty function to objective functions cannot provide effective information to lead the search. As a result, there is a large probability for SPEA2+$_{\text{pf}}$ to get into the local optimal solution in the early stage of the algorithm run.

Compared to SRA, CMOEA-c got better results. This is because the introduction of the assistant archive can help to explore the whole decision space. As we mentioned in Section 3, since the feasible regions are generally separated, once the constraint population converges to the optimal solution, the archive will keep search without considering the constraints. However, there is no such mechanism in SRA to help jump from one feasible region to another. Therefore, CMOEA-c can obtain better results.

4.3. Analysis of the Obtained Result. To further analyse the result obtained by CMOEA-c, we use Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [29] to pick the ideal solution as the final optimal result. Other approaches like knee point [30, 31] are also suitable for selecting satisfactory solutions. Specifically, the basic principle of TOPSIS is to sort solutions by calculating the

Table 1: The parameter values of all equipment used in HRES.

Equipment	Unit cost investment	Unit cost maintenance	Min capacity	Max capacity	Lifetime
Wind turbine	55	5	0	100	5
Photovoltaic	14	0.1	0	10000 (m^2)	25
Energy storage system	3	0.1	0	100	9
Diesel generator	2000	0.05	0	50	20
Power grid	—	—	—	10000 (kW)	—

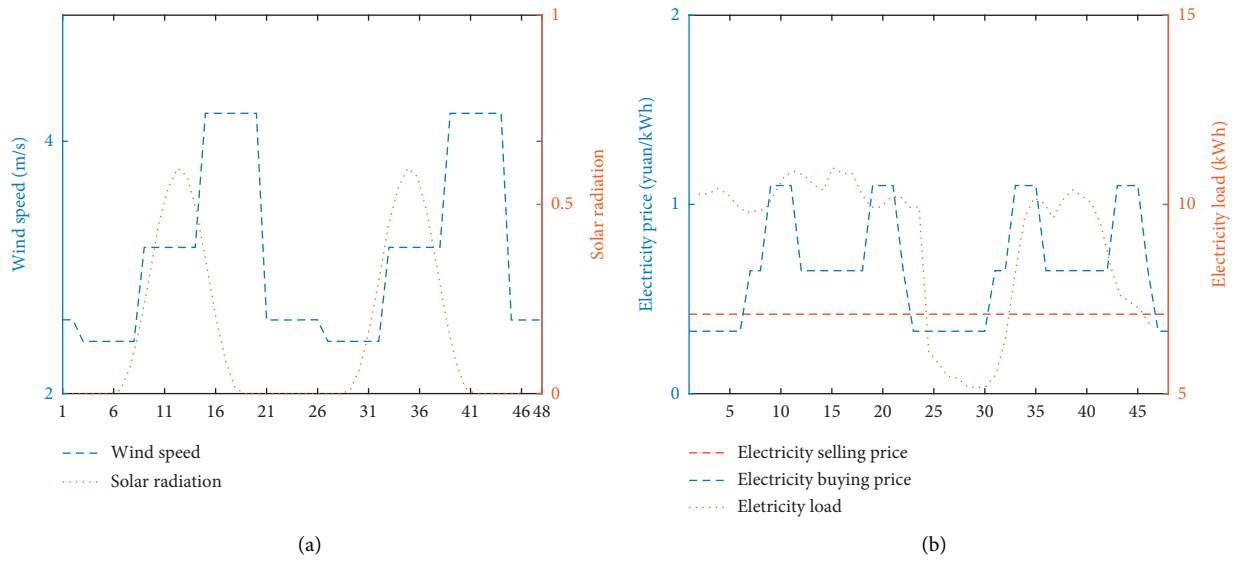


Figure 5: Running data of HRES in this study. (a) Wind speed and solar radiation. (b) User load and electricity price.

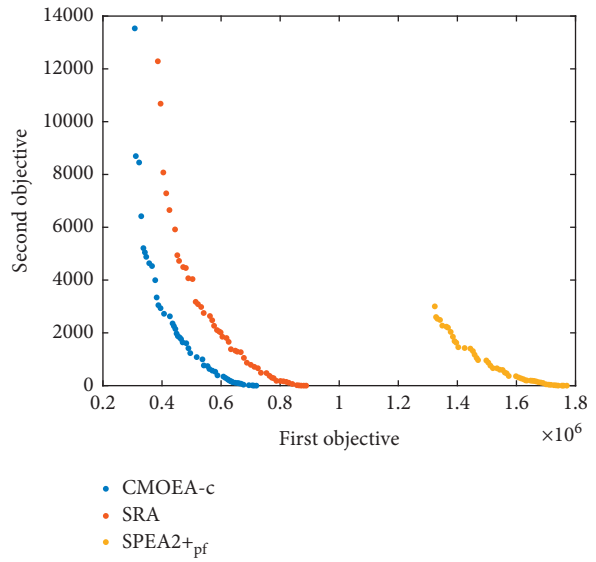


Figure 6: Pareto fronts obtained by CMOEA-c, SRA, and SPEA2+pf.

distances between the evaluating solution and the optimal and the worst solutions. If the evaluating solution is closest to the optimal solution and farthest from the worst solution, it is the best; otherwise, it is not an optimal solution. The running status of all components is presented in Figure 7.

As we can see from Figure 7(b), when the HRES is connected with the grid, it is more likely to sell electricity to the grid when the electricity purchasing price is high. This is because the system can minimize the running cost in this way. It is worth mentioning that wind turbine power generation is very close to electricity, and thus other components of the HRES are running with a low maintenance cost. In addition, when the HRES is isolated from the main grid, it needs to balance the power supply and demand all the time.

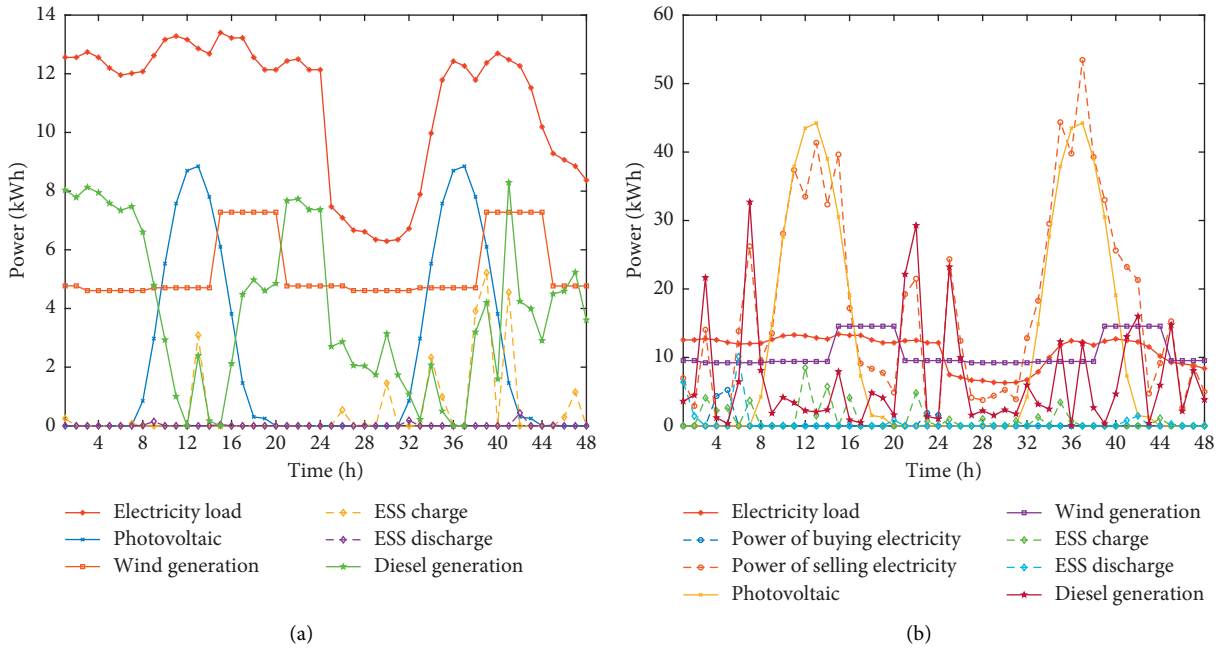


Figure 7: The equipment running status of HRES in this study. (a) Isolated microgrid. (b) Grid-connected microgrid.

To minimize the running cost of diesel generator and energy storage system, renewable energy generation (wind power and photovoltaic) is matched with the electricity load as much as possible as we can see from Figure 7(a). In conclusion, the proposed HRES model is effective which can be used to optimize the configuration in the designing stage. Moreover, our proposed CMOEA-c can effectively and efficiently solve HRES problems with nonlinear parts.

5. Conclusion

Microgrid is an effective tool to make better use of renewable energy. However, it is a tough task to determine the installed capacity of components, e.g., the number of wind turbine and the area of the photovoltaic cell. Thus, a hybrid renewable energy system (HRES) is proposed, which is generally nonlinear and large-scale. In addition, as more and more aspects are taken into consideration, an HRES problem needs to optimize more than one objective simultaneously. Therefore, traditional methods like linear programming are not suitable to solve this kind of problems.

In this paper, we first proposed the mathematical model of a hybrid renewable energy system, which has nonlinear parts. We consider two objectives to optimize, one is to minimize building and daily running cost, and another is to minimize the rate of lack of electricity. To solve this problem, we also proposed a coevolutionary multiobjective algorithm. Specifically, we use an archive to help explore the whole decision space. To examine the effectiveness of our proposed algorithm, several experiments are conducted, which show that the proposed algorithm can effectively find the Pareto optimal front of HRES problems.

We only consider a two-day (48 hours) energy management problem for HRES in this study. However, a longer period is necessary, e.g., one year. Such problems are large-scale which have more than 10000 decision variables. So far, it is really difficult to solve this issue with evolutionary algorithms [32, 33], which is our future work. On the other hand, the decision variables of the HRES problem studied in this work are highly correlated. Specifically, the selecting state can significantly affect other decision variables. For such a mixed-integer problem, we need to develop a more effective algorithm.

Data Availability

The research data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61773390), the Young Talents of Hunan (2018RS3081), and the Scientific Research Project of National University of Defense Technology (ZZKY-ZX-11-04 and ZK18-02-09).

References

[1] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: a case study: the brooklyn microgrid," *Applied Energy*, vol. 210, pp. 870–880, 2018.

[2] V. Khare, S. Nema, and P. Baredar, "Solar-wind hybrid renewable energy system: a review," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 23–33, 2016.

[3] M. Ming, R. Wang, Y. Zha, and T. Zhang, "Multi-objective optimization of hybrid renewable energy system using an enhanced multi-objective evolutionary algorithm," *Energies*, vol. 10, no. 5, p. 674, 2017.

[4] R. Hosseinalizadeh, H. Shakouri G, M. S. Amalnick, and P. Taghipour, "Economic sizing of a hybrid (PV-WT-FC) renewable energy system (HRES) for stand-alone usages by an optimization-simulation model: case study of Iran," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 139–150, 2016.

[5] M. Amer, A. Namaane, and N. K. M'Sirdi, "Optimization of hybrid renewable energy systems (HRES) using PSO for cost reduction," *Energy Procedia*, vol. 42, pp. 318–327, 2013.

[6] S. Vigerske and A. Gleixner, "SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework," *Optimization Methods and Software*, vol. 33, no. 3, pp. 563–593, 2018.

[7] R. Wang, G. Li, M. Ming, G. Wu, and L. Wang, "An efficient multi-objective model and algorithm for sizing a stand-alone hybrid renewable energy system," *Energy*, vol. 141, pp. 2288–2299, 2017.

[8] W. Li, T. Zhang, and R. Wang, "Energy management model of charging station micro-grid considering random arrival of electric vehicles," in *Proceedings of the 2018 IEEE International Conference on Energy Internet (ICEI)*, pp. 29–34, Beijing, China, May 2018.

[9] W. Li, "Knee point-guided multiobjective optimization algorithm for microgrid dynamic energy management," *Complexity*, vol. 2020, Article ID 8877008, 2020.

[10] C. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[13] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, pp. 832–842, Birmingham, UK, September 2004.

[14] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] R. Wang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2016.

[16] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: a comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

[17] Q. Zhang, G. Pang, and G. Wang, "A novel sequential three-way decisions model based on penalty function," *Knowledge-Based Systems*, vol. 192, Article ID 105350, 2020.

[18] A. Panda and S. Pani, "A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems," *Applied Soft Computing*, vol. 46, pp. 344–360, 2016.

[19] W. Li, R. Wang, T. Zhang, M. Ming, and H. Lei, "Multi-scenario microgrid optimization using an evolutionary multi-objective algorithm," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100570, 2019.

[20] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.

[21] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2021.

[22] A. Salhi, G. Alsoufi, and X. Yang, "An evolutionary approach to a combined mixed integer programming model of seaside operations as arise in container ports," *Annals of Operations Research*, vol. 272, no. 1-2, pp. 69–98, 2019.

[23] K. L. Sadowski, M. C. van der Meer, N. H. Luong et al., "Exploring trade-offs between target coverage, healthy tissue sparing, and the placement of catheters in hdr brachytherapy for prostate cancer using a novel multi-objective model-based mixed-integer evolutionary algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1224–1231, Berlin, Germany, July 2017.

[24] H.-Z. Wang, G.-Q. Li, G.-B. Wang, J.-C. Peng, H. Jiang, and Y.-T. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Applied Energy*, vol. 188, pp. 56–70, 2017.

[25] R. Wang, W. Ma, M. Tan et al., "Preference-inspired co-evolutionary algorithm with active diversity strategy for multi-objective multi-modal optimization," *Information Sciences*, vol. 546, pp. 1148–1165.

[26] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2018.

[27] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2012.

[28] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2," in *International Conference on Parallel Problem Solving from Nature*, pp. 742–751, Birmingham, UK, September 2004.

[29] D. Niu, Y. Li, S. Dai et al., "Sustainability evaluation of power grid construction projects using improved topsis and least square support vector machine with modified fly optimization algorithm," *Sustainability*, vol. 10, no. 1, p. 231, 2018.

[30] W. Li, R. Wang, T. Zhang, M. Ming, and K. Li, "Reinvestigation of evolutionary many-objective optimization: focus on the pareto knee front," *Information Sciences*, vol. 522, pp. 193–213, 2020.

[31] W. Li, T. Zhang, R. Wang, B. Wang, Y. Song, and X. Li, "A knee-point driven multi-objective evolutionary algorithm for flexible job shop scheduling," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1716–1722, Xiamen, China, December 2019.

[32] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.

[33] R. Cheng, Y. Jin, and M. Olhofer, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2016.

Hindawi
Complexity
Volume 2021, Article ID 6627081, 14 pages
https://doi.org/10.1155/2021/6627081

WILEY | Hindawi

Research Article

Two-Agent Single Machine Order Acceptance Scheduling Problem to Maximize Net Revenue

Jiaji Li,[1] **Yuvraj Gajpal**,[1] **Amit Kumar Bhardwaj**,[2] **Huangen Chen**,[3] **and Yuanyuan Liu**[4]

[1]*Department of Supply Chain Management, I.H. Asper School of Business, University of Manitoba, Winnipeg, MB R3T 5V4, Canada*
[2]*L.M Thapar School of Management, Thapar Institute of Engineering & Technology, Dera Bassi Campus, Patiala, Punjab, India*
[3]*School of Business Administration, Southwestern University of Finance and Economics, Chengdu, China*
[4]*School of International Business, Southwestern University of Finance and Economics, Chengdu, China*

Correspondence should be addressed to Yuanyuan Liu; liuyuanyuan@swufe.edu.cn

Received 1 November 2020; Revised 17 January 2021; Accepted 20 January 2021; Published 8 February 2021

Academic Editor: Shangce Gao

The paper considers two-agent order acceptance scheduling problems with different scheduling criteria. Two agents have a set of jobs to be processed by a single machine. The processing time and due date of each job are known in advance. In the order accepting scheduling problem, jobs are allowed to be rejected. The objective of the problem is to maximize the net revenue while keeping the weighted number of tardy jobs for the second agent within a predetermined value. A mixed-integer linear programming (MILP) formulation is provided to obtain the optimal solution. The problem is considered as an *NP-hard* problem. Therefore, MILP can be used to solve small problem instances optimally. To solve the problem instances with realistic size, heuristic and metaheuristic algorithms have been proposed. A heuristic method is used to determine and secure a quick solution while the metaheuristic based on particle swarm optimization (PSO) is designed to obtain the near-optimal solution. A numerical experiment is piloted and conducted on the benchmark instances that could be obtained from the literature. The performances of the proposed algorithms are tested through numerical experiments. The proposed PSO can obtain the solution within 0.1% of the optimal solution for problem instances up to 60 jobs. The performance of the proposed PSO is found to be significantly better than the performance of the heuristic.

1. Introduction

Order acceptance scheduling has been studied by researchers for the last few decades. In many cases, the number of orders accepted by a firm is not necessarily positively related to its profit, especially, when the capacity is bounded. Firms need to reject some of the orders to maximize their profit when the associated costs for some of the orders exceed their revenues. Accepting orders without a sensible and logical consideration might have a directly proportionate impact on the increase in workload. When the workload is excessively heavy, compared to the capacity of the firm, it might incur late deliveries of orders, therefore decreasing the level of customer satisfaction or even losing important customers. Hence, the study having order acceptance as its core focus has grabbed the attention of many researchers.

In the field of scheduling, researchers considered order acceptance in the last few decades. In the scheduling problem, decision-makers need to decide which jobs to be processed and what will be the sequence of the accepted jobs (Li et al. [1]). There is a trade-off between the revenues and penalty function while deciding acceptance or rejection of the jobs. There are always some penalty functions available,

related to the due dates, such as lateness and tardiness, which are worth looking into, depending on the scenario requirement. The order acceptance scheduling problem aims to maximize the profit gained by the revenues of accepted orders minus the specific penalty function.

So far, many researchers have tried to apply the study to the order acceptance problems, testing its usefulness in multiagent settings. In the multiagent scheduling problem, the jobs of two agents with conflicting objectives are processed by common resources (Gajpal and Li [2]; Li et al. [3]). The objectives of the order acceptance scheduling problem and multiagent scheduling problem are similar. Both refer to the construction of the settings for more explicitly specified conditions. The paper of Reisi-Nafchi and Moslehi [4] depicts a more practical application portraying the combination of the order acceptance with multiagent scheduling problems. In their research, they formulated a mathematical model for their problem and proposed a hybrid genetic algorithm to solve this problem.

Inspired by the work of Reisi-Nafchi and Moslehi [4], this paper extends their study, highlighting the following three perspectives: (1) propose a new variant of the two-agent order acceptance scheduling problem; (2) propose a mathematical model for the new variant of the problem; and (3) design algorithms that can be used to solve both versions of the problem. This paper focuses on two problems named as lateness penalty problem (*LPP*) and the tardiness penalty problem (*TPP*). The *LPP* problem is proposed by Reisi-Nafchi and Moslehi [4]; however, the *TPP* problem has been introduced in this paper, for the very first time. Both problems are the two-agent single machine order acceptance scheduling problems but their objective functions are different. The objective function of both problems considers how to maximize the net revenue of the first agent, subject to an upper bound on the weighted number of tardy jobs for the second agent. The net revenue is calculated as revenue from the accepted jobs subtracted by the penalty function. Penalty functions take the punishment by decreasing the objective function value when job processes after their due dates. The penalty value of the *LPP* problem is the weighted lateness of the first agent while the penalty value of the *TPP* problem is the weighted tardiness of the first agent.

To solve these problems, one metaheuristic and one heuristic algorithm have been introduced. A mathematical model based on MILP formulation is developed to find the optimal solution for the problem. The MILP formulation can solve only small size problem instances because the CPU time for solving MILP formulation increases exponentially with the size of the problem. To solve the problem instances with realistic size, heuristic and metaheuristic algorithms have been proposed.

The scheduling problem considered in this paper could be applied to the following scenario. A restaurant provides service to two kinds of customers: (1) customers who dine in the restaurant and (2) customers who order take-out services. Jobs in this scheduling problem are the dishes to be served to the customers. The two types of customers are treated as two agents. During the rush hours, the restaurant could not accept all the orders. Thus, it needs to reject some of the orders based on the profitability of the orders. The objective of the restaurant is to maximize the net revenue coming from all the orders while the weighted number of tardy jobs from the take-out customers could not exceed the upper bound. Also, the problem considered in this paper has many practical applications in the manufacturing industry as well as in the service industry. One of the applications that fall under this paper's criteria is the cloud-computing environment (Bhardwaj et al. [5]). In the cloud-computing environment, the service provider accepts the order of processing jobs from different users along with his own jobs. The service provider can be considered as an agent A and all other users combined can be considered as an agent B. The objective of the cloud-computing service provider is to maximize the net revenue from all the accepted jobs, while the weighted number of tardy jobs from all other users is kept within a predefined limit set by the management.

The remainder of the paper is organized as follows. Related work is reviewed in Section 2. A detailed description of the problems and the problem formulation is presented in Sections 3 and 4. The proposed algorithms are described in Section 5. Numerical results performed in Section 6 are used for comparing the effectiveness of the proposed algorithms. Eventually, Section 7 presents a summary of this paper.

2. Literature Review

Guerrero and Kern [6] are the pioneers in the study of order acceptance problems. They pointed out the importance of selecting a proper number of orders instead of accepting orders without any sensible cogitation. Pourbabai [7] first developed a mathematical model to help the manufacturers select the orders among all the orders. Slotnick and Morton [8] worked on an order acceptance problem in a single machine environment, of which the objective is the revenues minus weighted lateness penalties. They proposed an exact algorithm (branch and bound) and two heuristics to solve the problem. Ghosh [9] extended the work of Slotnick and Morton [8] by providing the NP-hardness proof of the problem. Also, they provided two pseudopolynomial algorithms to solve the problem, along with a fully polynomial-time approximation scheme. Lewis and Slotnick [10] worked on a multiperiod order acceptance scheduling problem in which decisions on order acceptance are made through different periods. They assume that job rejection leads to the loss of customers, and the profitability of decisions is scrutinized and assessed over each period. They proposed a dynamic programming algorithm and some heuristics to solve the problem. Slotnick and Morton [11] considered another order acceptance scheduling problem by replacing the penalty function with weighted tardiness. Rom and Slotnick [12] applied a genetic algorithm to solve the order acceptance scheduling problem with tardiness revenues.

Many researchers added variants to the order acceptance scheduling problem. Charnsirisakskul et al. [13] studied a single machine order acceptance problem with job preemption. Oğuz et al. [14] considered sequence-dependent setup times in their problem and arbitrarily assigned deadlines to the orders. Zhong et al. [15] added the variant of

machine availability to order acceptance scheduling problems, in which jobs could only be processed during specific time intervals due to the availability of the machines. When the workload is heavier than the capacity, manufacturers should decide either to reject or to outsource some of the orders. Xiao et al. [16] worked on an order acceptance scheduling problem in a permutation flow shop environment. They formulated their problem into an integer-programming problem and applied simulated annealing based on partial optimization to solve the problem. Lei and Guo [17] considered an order acceptance problem in a workshop environment. They contrived, putting together this particular problem and blending it into a mixed-integer programming problem, and applied the parallel neighborhood search algorithm to it.

There are some studies on order acceptance scheduling with bicriteria objectives. Lei and Guo [17] considered the objective of maximizing total net revenue and minimization of makespan simultaneously. Ou and Zhong [18] worked on a bicriteria order acceptance scheduling problem, with upper bounds on the number of the rejected job as well as the total processing time of accepted jobs. Noroozi et al. [19] considered order accepting criteria in a third-party logistic distribution system. They analyzed the trade-off among accepted orders revenue, delivery cost, and tardiness penalty in integrated production-distribution to maximize benefit. Silva et al. [20] considered an order acceptance scheduling problem with sequence-dependent setup time to maximize profit. They solved the problem using the Lagrangian relaxation and column generation-based branch and bound method. Sarvestani et al. [21] addressed an integrated order acceptance, supplier selection, and scheduling problem to maximize profit. They used genetic algorithm (GA) and variable neighborhood search (VNS) to solve the problem. Li and Ventura [22] considered a single-agent single machine scheduling problem with order acceptance criteria to maximum profit. The profit function considers the revenue minus the tardiness penalty. Recently, order acceptance criteria are combined with different settings/environments such as parallel machine setting (Wang and Ye [23]; Wu et. al [24]; Palakiti et al. [25]); carbon reduction environment (Chen et al. [26]); cluster supply chain environment (Li et al. [27]); production system (Wang et al. [28]; Yavari et al. [29]); and dynamic environment (Li et al. [30]; Melchiors et al. [31]). The increasing number of research articles in the order accepting criteria under different settings shows the importance and popularity of order acceptance scheduling problems.

The article by Reisi-Nafchi and Moslehi [4] is the closely related model considered in this paper. They integrated the two-agent scheduling situation into the order acceptance scheduling problem to maximize the net revenues from the first agent, subject to an upper bound of a weighted number of tardy jobs from the second agent. The net revenue is considered as a revenue of accepted orders with a penalty of total weighted lateness of accepted jobs. They provided an integer-programming formulation and hybrid genetic algorithm to solve the problem. This paper extends the work of Reisi-Nafchi and Moslehi [4] by introducing a new variant of the order acceptance criteria in the "two agents scheduling" setting. The literature review also stipulates the common trend of using metaheuristics to solve the scheduling problem. This observation has motivated us to develop and incorporate a metaheuristic algorithm to solve our problem.

3. Problem Definition and Notations

This paper considers a single machine scheduling problem with two agents, namely, agent A and agent B. Each of the agents has a job set n^x to be processed in a single machine. The processing time p_i, weight w_i, due date D_i , and revenue R_i associated with the job i are known. All the jobs are available at time zero (i.e., the release time is zero). A single machine processes all the jobs and at a time only one job is processed. Once a machine has started processing a job, it will continue running on that job until the job is completed. The problem seeks to determine the sequence of jobs σ. For a given sequence σ, the complete time $C_i(\sigma)$ of accepted job i can be calculated by the summation of processing time of all the jobs (including job i) sequenced before job i. The lateness value L_i of the job is defined as a difference between completion time and due date (i.e., $L_i = C_i(\sigma) - D_i$). The lateness value is positive if the job is completed after its due date. On the other hand, the lateness value is negative if the job is completed before the due date. Slotnick and Morton [8] introduced the lateness penalty to capture the penalty and reward in one function. Finishing a job late incurs a penalty in terms of loss of goodwill while finishing the job early incurs an appreciation from the customer. However, in many situations, only late completion incurs a penalty and the early completion does not bring any appreciation. This situation is captured in a tardiness value which only measures the delay in the completion time of a job (i.e., $T_i = \text{Max}\{0, C_i(\sigma) - D_i\}$). The existing literature studies the order acceptance scheduling problem in which lateness penalizes revenue obtained from the accepted jobs. This paper introduces a new order acceptance scheduling problem variant in which tardiness penalizes revenue received from the accepted jobs. The summary of notations used in this paper is summarized as follows:

(i) X: set of agents $X = \{A, B\}$

(ii) n_x: the number of jobs for agent $x \in X$

(iii) n: total number of jobs, $n = n_A + n_B$

(iv) N: set of all n jobs, $N = \{1, 2, \ldots, n\}$

(v) N_A: set of A jobs, consisting of n_A jobs, $N_A = \{1, 2, \ldots, n_A\}$

(vi) N_B: set of B jobs, consisting of n_B jobs, $N_B = \{n_A + 1, n_A + 2, \ldots, n\}$

(vii) p_i: the processing time of job i

(viii) R_i: revenue of job i

(ix) w_i: weight of job i

(x) D_i: due date of job i

(xi) σ: ordered set of jobs already scheduled

(xii) $C_i(\sigma)$: completion time of accepted job i under sequence σ.

(xiii) $f^x(\sigma)$: objective function of agent $x \in X$ under sequence σ.

(xiv) Y_i: binary decision variable taking value 1 if job i is accepted

(xv) U_i: binary decision variable taking value 1 if job i is tardy

(xvi) Z_{ij}: binary decision variable taking value 1 if job i precedes job j

(xvii) T_i: decision variable representing tardiness of the job i

(xviii) L_i: decision variable representing lateness of the job i

(xix) (xx) Q: the upper bound, a constant numberbig M: a large number

This paper showcases an attempt to maximize the net revenue subject to an upper bound of agent B. We consider two versions of the order acceptance two-agent scheduling problems, the *LPP* problem and the *TPP* problem. The objective of the *LPP* problem is to maximize the net revenue from all the accepted orders subtracted by the total weighted lateness from the accepted jobs of agent A, subject to an upper bound of the total weighted number of tardy jobs from agent B. In the scheduling terminology, this problem can be represented as $1|OA, \sum_{i\in N_B} w_i U_i Y_i \le Q|\sum_{i\in N} R_i Y_i - \sum_{i\in N_A} w_i L_i Y_i$. The term OA is used for order acceptance. The objective of the *TPP* problem is to maximize the net revenue from all the accepted orders subtracted by the total weighted tardiness from the accepted jobs of agent A, subject to an upper bound of the total weighted number of tardy jobs from agent B. This problem can be represented as $1|OA, \sum_{i\in N_B} w_i U_i Y_i \le Q|\sum_{i\in N} R_i Y_i - \sum_{i\in N_A} w_i T_i Y_i$.

The computational complexity of the *LPP* problem is proved to be NP-hard by Reisi-Nafchi and Moslehi [4]. They showed that the *LPP* problem is a special case $1|\sum_{i\in N_B} w_i U_i \le Q|\sum_{i\in N_A} w_i L_i$; therefore, the *LPP* problem is also an NP-hard problem. The *TPP* problem reduces to $1|\sum_{i\in N_B} w_i T_i$ when Q is set to infinite. Since $1|\sum_{i\in N_B} w_i T_i$ is *NP-hard*, the *TPP* problem is also an NP-hard problem.

4. Problem Formulation

Reisi-Nafchi and Moslehi [4] formulated the *LPP* problem into an integer-programming mathematical model. Based on their formulation, we have brought forward an evolved version of the paper, which is developed as an integer-programming model for the *TPP* problem.

When formulating the mathematical model for the *TPP* problem, a new binary variable Z_{ij} is introduced, which denotes the relative position between jobs i and j. The binary variable Z_{ij} takes the value of 1 if job i precedes job j, otherwise zero. It is significant to note here that the precedence here is not the immediate precedence (i.e., Z_{ij} takes the value of 1; this does not mean that job i and job j are adjacent to each other). Using the above notations, the *TPP* problem can be formulated as follows:

$$\text{Max} \sum_{i\in N} R_i Y_i - \sum_{i\in N_A} w_i T_i, \tag{1}$$

$$\text{subject to } Y_i + Y_j \le 1 + Z_{ij} + Z_{ji}, \quad i, j \in N,\ i < j, \tag{2}$$

$$Y_i + Y_j \ge 2\left(Z_{ij} + Z_{ji}\right), \quad i, j \in N,\ i < j, \tag{3}$$

$$Z_{ik} + Z_{kj} + Z_{jk} \le 2, \quad i, j, k \in N,\ i \ne j,\ j \ne k,\ i \ne k, \tag{4}$$

$$T_i \ge \sum_{j\in N} p_j Z_{ji} + p_i Y_i - D_i, \quad i \in N_A, \tag{5}$$

$$T_i \ge 0, \quad i \in N_A, \tag{6}$$

$$\sum_{j\in N} p_j Z_{ji} + p_i Y_i \le D_i Y_i + \text{Big}\, M \times U_i, \quad i \in N_B, \tag{7}$$

$$\sum_{j\in N_B} w_j U_j \le Q, \tag{8}$$

$$U_i \in \{0, 1\}, \quad i \in N_B, \tag{9}$$

$$Y_i \in \{0, 1\}, \quad i \in N, \tag{10}$$

$$Z_{ij} \in \{0, 1\}, \quad i, j \in N. \tag{11}$$

Equation (1) represents the objective function of the *TPP* problem, which is the total revenue of all the accepted orders penalized by the weighted tardiness of agent A orders. Constraints (2)–(4) determines the relative positions of jobs. Constraint (2) requires that a job i must precede another job j if both jobs are accepted. Constraint (3) sets the value of Z_{ij} or Z_{ji} to zero when job i or j is rejected, preventing the situation that Z_{ij} and Z_{ji} both are equal to zero. Constraint (4) ensures that there are no job loops in the sequence. Job loop represents a situation in which job i precedes job j, job j precedes job k, and job k precedes job i, which is logically wrong. Constraints (5) and (6) determine the tardiness value for job i. Constraint (7) determines the value of U_i. Constraint (8) ensures that the upper bound of the weighted number of tardy jobs from agent B is not exceeding. Constraints (9)–(11) impose the binary values for binary decision variables U_i, Y_i, and Z_{ij}.

Some of the optimal solution property of the *LPP* problem also holds true for *TPP* problem. The *TPP* problem satisfies the following three lemmas for which proof can be found in the work of Reisi-Nafchi and Moslehi [4].

Lemma 1. *There exists an optimal sequence, in which the agent B tardy accepted orders are sequenced arbitrarily at the end of the sequence.*

Lemma 2. *There exists an optimal sequence, in which the global arrangement of the agent B nontardy accepted orders follows the earliest due date (EDD) order and they are placed at the last possible positions before their due dates.*

Lemma 3. *There exists an optimal sequence, in which the adjacent orders of agent A follow the weighted shortest processing time (WSPT) order.*

5. Proposed Algorithms

In this section, algorithms including a heuristic and PSO are the proposed approaches to resolve the *LPP* and *TPP* problems.

5.1. Heuristic Algorithm. The proposed heuristic is based on the *WSPT* and *EDD* rule. In the WSPT rule, jobs are sequenced in nondecreasing order of the p_i/w_i ratio. In the EDD rule, jobs are sequenced in nondecreasing order of their due dates D_i. The WSPT rule minimizes the total weighted completion time while the EDD rule minimizes the total tardiness objective for a single machine and a single-agent setting.

The heuristic first creates partial schedules σ_A and σ_B for A and B types of jobs according to WSPT and EDD rules, respectively. The sequence σ_A is modified by removing unprofitable jobs. The sequence σ_B is divided into two subsequences σ_B^E and σ_B^L for an early and late set of the jobs. The final sequence σ is created by first sequencing σ_A , then σ_B^E , and finally σ_B^L. If the resultant sequence σ is infeasible, then the jobs from the sequence σ are removed one by one to make the solution feasible. The pseudocode of the proposed heuristic algorithm is presented as follows (Algorithm 1).

The computational complexity of the proposed algorithm can be described as follows. The initial solution generation has four separate subprocesses: (1) sequence of A jobs with complexity $O(n_1^2)$; (2) sequence of B jobs with complexity $O(n_2^2)$; (3) split of the sequence of B jobs with complexity $O(n_2)$; and (4) resequencing of late B jobs with complexity $O(n_2^2)$. Thus, the overall complexity of the initial solution process is $O(n_1^2) + O(n_2^2) + O(n_2) + O(n_2^2) = O(n_1^2) + O(n_2^2)$. The feasibility phase requires the maximum removal of n jobs and after each removal, it requires objective function calculation which can be performed in $O(n)$ time. Thus, the feasibility phase can be performed in $O(n^2)$ time. Therefore, the overall complexity of the heuristic algorithm is $O(n_1^2) + O(n_2^2) + O(n^2) = O(n^2)$.

5.2. Particle Swarm Optimization. Particle swarm optimization (PSO) is a metaheuristic technique for solving NP-hard problems. PSO is derived from the social behavior of fish and birds. In this algorithm, individual solutions are updated by the other members of the group. Although PSO applies to many scheduling problems, there are some challenges while applying it to solve a specific problem. The main challenge for this algorithm is to design a solution representation mechanism [32, 33] for the particular problem. In other words, this refers to the mechanism, which establishes the rules to interpret the data of particles into an actual schedule. Additional efforts are required to make it pertinent to the order acceptance scheduling problems because two subproblems are residing within it. While the first subproblem determines the set of accepted jobs, the objective of the second subproblem is to find the sequence for the accepted jobs. Therefore, a PSO algorithm containing parallel parameters of position values and velocities has been proposed. It is difficult for the classical PSO with only one group of parameters to solve the problems in this paper; two groups of position values and velocities are embedded in the algorithm. The first group focuses on the selection of accepted jobs, and the second group focuses on finding the sequence of the accepted jobs. Each particle in PSO has n dimensions representing n jobs. Let X_j^1 and X_j^2 represent the position value vector related to the order acceptance decision making and the job sequencing, respectively. The following are the notations that we have employed to describe the proposed PSO algorithm:

(i) m: number of particles

(ii) k: iterations index

(iii) K: maximum number of iterations

(iv) x_{ij}^1: position value of i^{th} particle in j^{th} dimension for order acceptance criteria

(v) X_i^1: position vector of i^{th} particle for order acceptance criteria

(vi) v_{ij}^1: velocity value of i^{th} particle in j^{th} dimension for order acceptance criteria

(vii) V_i^1: velocity vector of i^{th} particle for order acceptance criteria

(viii) x_{ij}^2: position value of i^{th} particle in j^{th} dimension for job sequencing criteria

(ix) X_i^2: position vector of i^{th} particle for job sequencing criteria

(x) v_{ij}^2: velocity value of i^{th} particle in j^{th} dimension for job sequencing criteria

(xi) V_i^2: velocity vector of i^{th} particle for job sequencing criteria

(xii) (xiii) pb_{ij}: personal best position value of i^{th} particle in j^{th} dimension for job sequencing criteriagb$_j$: global best position value in j^{th} dimension for job sequencing criteria

(xiv) S_i^{PB}: personal best solution vector for i^{th} particle

(xv) S^G: global best solution

(xvi) w: inertia weight

(xvii) c_1, c_2: social cognitive numbers

The pseudocode of the proposed PSO algorithm is as follows (Algorithm 2).

5.2.1. Parameter Initialization. At the beginning, parameters are generated using the following equations:

$$\begin{aligned} x_{ij}^1 &= X_{\min}^1 + \left(X_{\max}^1 - X_{\min}^1\right) \times U(0,1), \\ v_{ij}^1 &= V_{\min}^1 + \left(V_{\max}^1 - V_{\min}^1\right) \times U(0,1), \\ x_{ij}^2 &= X_{\min}^2 + \left(X_{\max}^2 - X_{\min}^2\right) \times U(0,1), \\ v_{ij}^2 &= V_{\min}^2 + \left(V_{\max}^2 - V_{\min}^2\right) \times U(0,1), \end{aligned} \tag{12}$$

where $X_{\max}^1, V_{\max}^1, X_{\max}^2, V_{\max}^2$ are equal to 4; $X_{\min}^1, V_{\min}^1, X_{\min}^2, V_{\min}^2$ are equal to -4. $U(0,1)$ is a random number within the range of [0, 1]. Therefore, all these parameters remain within the range of $[-4, 4]$ and the same is maintained throughout the algorithm.

5.2.2. Sequence Generator. As mentioned above, the sequence generating process in the proposed PSO consists of two steps. The first step is to decide and govern the accepted jobs and the second step decides the sequence of the accepted jobs. Job j in particle i is accepted when x_{ij}^1 is greater than 0; otherwise, it is rejected. Jobs are sequenced according to the minimum position value (MPV) rule, which specifies that the jobs are ordered in nondecreasing order of their position values x_{ij}^2.

5.2.3. Updating Parameters. Parameters are updated at the end of each iteration. Before updating the parameter, the personal best solution S_i^{pb} and the personal best position value pb_{ij} for each particle are updated. If the new solution σ_i is found to be better than S_i^{pb}, then the personal best solution S_i^{pb} is replaced by the current solution $\sigma_i$and the personal best position value pb_{ij} is replaced by the position value x_{ij}^2. Similarly, the global best solution and the global best position values are updated. Two different ways are applied for updating job sequencing and order accepting criteria. The order acceptance parameters as updated as follows:

$$\begin{aligned} v_{ij}^1 &= w \times v_{ij}^1 + c_1 \times r_1 \times z_1 + c_2 \times r_2 \times z_2, \\ x_{ij}^1 &= x_{ij}^1 + v_{ij}^1. \end{aligned} \tag{13}$$

Parameters of z_1 and z_2 are variables representing the status of job j. Variable z_1 takes the value of 1 when job j in i^{th} particle is accepted in personal best solution; otherwise, it takes the value of -1. Likewise, the value of the variable z_2 is determined by whether the job is accepted in the global best solution or not. Parameters of c_1 and c_2 are the social coefficients quantifying the extent to which an individual relies on its experiment (i.e., to its own personal best or to the global best). Inertia w is the resistance of any physical object to any change in the state of motion. It controls the velocity and direction of all the particles. Variables of r_1 and r_2 are random numbers within the range of [0, 1]. In this paper, we set $c_1 = 0.9, c_2 = 0.9, w = 0.9$. The job sequencing parameters are updated as follows:

$$\begin{aligned} v_{ij}^2 &= w \times v_{ij}^2 + c_1 \times r_1 \times \left(\text{pb}_{ij} - x_{ij}^2\right) + c_2 \times r_2 \times \left(\text{gb}_{ij} - x_{ij}^2\right), \\ x_{ij}^2 &= x_{ij}^2 + v_{ij}^2. \end{aligned} \tag{14}$$

5.2.4. Local Search. There are two types of local search schemes proposed to improve the solution of each particle. These two local search schemes are named as an *accepted job insertion local search (AJILS)* and a *rejected job insertion local search (RJILS)*. The *AJILS* tries to improve the solution by repositioning the accepted jobs to a different position. At first, the job is removed from its original position, and then it was reinserted in all the other positions. The best-inserted position is selected for the insertion of the job and the resultant solution is used for further evaluation. All the accepted jobs are selected one by one for possible solution improvement. The *RJILS* attempts to insert the rejected jobs into the solution. The rejected jobs are inserted in the best feasible position if it improves the objective function value. All the rejected jobs are singled out one by one for possible solution improvement.

5.2.5. Normalization of Parameters. The absolute values of parameters tend to be very large after many iterations, enabling the solutions to get stuck in the local optimum. Thus, the parameters are normalized after each iteration. The parameters exceeding the upper limit are forced to take a value of 4, and the parameters that are less than the lower limit are forced to take the value of -4.

5.2.6. Computational Complexity of the PSO. The three main components of the PSO algorithm are (1) sequence generation step, (2) local search method, and (3) parameter updating step. In the sequence generation step, determining the set of accepted and rejected jobs can be performed in $O(n)$ time while the sequencing operation can be performed in $O(n^2)$ time. Thus, the overall complexity of the sequence generation step is $O(n^2)$. The local search uses the two schemes with the complexity of (n^2) for each scheme. Thus, the overall complexity of the local search scheme is $O(n^2)$. The parameter updating can be performed in $O(n)$ time. Thus, the complexity of one iteration of PSO is $O(n^2) + O(n^2) + O(n) = O(n^2)$. The PSO performs K iterations, which brings the overall complexity of the PSO to be $O(Kn^2)$.

6. Numerical Results

This section evaluates the performance of the proposed algorithms, assessing their effectiveness and application utility on two problems. Numerical analysis is provided based on the solution and CPU time of the proposed algorithms. The proposed algorithms are coded in the C++ programming language. The numerical experiments are

```
(1) /* Initial Solution */
(2) Create a sequence σ_A for A jobs according to the WSPT rule
(3) for k = 1 to n_1 do
(4)    Select job i = σ_A(k)
(5)    if R_i < w_i × (C_i(σ) − D_i) then
(6)       Remove job i from sequence σ_A
(7)    end if
(8) end for
(9) Create a sequence σ_B for B jobs according to the EDD rule
(10) Create null sequences σ_B^E = {φ} and σ_B^L = {φ}
(11) for k = 1 to n_2 do
(12)    Select job i = σ_B(k)
(13)    if C_i(σ_A) + p_i <= D_i then
(14)       append job i in sequence σ_B^E
(15)    else
(16)       append job i in sequence σ_B^L
(17)    end if
(18) end for
(19) Update σ_B^L by arranging jobs in nondecreasing order of R_i/w_i ratio
(20) Create initial sequence σ = {σ_A ∪ σ_B^E ∪ σ_B^L}
(21) /* Feasibility phase */
(22)    U(σ) ⟵ f^B(σ) //Calculate the weighted number of tardy jobs for σ
(23) for k = 1 to n do
(24)    if U(σ) ≤ Q then
(25)       break k loop
(26) end if
(27)    σ ⟵ Update sequence by removing first job
(28)    U(σ) ⟵ f^B(σ) //Calculate the weighted number of tardy jobs for σ
(29) end for
(30) return solution σ
```

Algorithm 1: Heuristic algorithm.

performed in a computer environment with AMD Opteron 2.3 GHz with 16 GB of RAM on Unix OS. The linear-programming-based mathematical model proposed for the *TPP* problem is solved by AMPL software with CPLEX solver, in the computer system of iMac desktop with 3.3 GHz 8 GB RAM.

There are three sections in this section. The first section explains the settings of the instances used in this paper. The second section aims to test the robustness of proposed algorithms by comparing them with the existing solution of Reisi-Nafchi and Moslehi [4]. The third section evaluates the proposed algorithms on the *TPP* problem. The following are notations used to report and evaluate the numerical results:

(i) *CPLEX*: value of objective function generated by the mathematical model

(ii) *ABS*: absolute value of objective function generated by algorithms

(iii) ABS_{best}: *ABS* value generated by the algorithm with the best performance

(iv) *APD*: absolute percentage of deviation, APD = ((ABS − CPLEX)/CPLEX ∗ 100%)

(v) *RPD*: relative percentage of deviation, RPD = $((ABS - ABS_{best})/ABS_{best} * 100\%)$

(vi) *CPU*: running time of the proposed algorithm

Additionally, *Avg*, *Min*, and *Max* are used for representing average, minimal, maximal values of corresponding results, respectively. We also performed a paired t-test to compare the performance of the proposed algorithms at the 95% significance level. In the paired t-test, the population mean of the two methods is compared. Let $\mu_D = \mu_1 - \mu_2$ denote the true average difference between the average solution values of the comparing algorithms. Here, μ_1 and μ_2 denote the population mean of the comparing algorithms, respectively. The null hypothesis is given by H_0: $\mu_D = 0$, which states that there is no difference between the average solution value of the algorithms. The alternative hypothesis is given by H_1: $\mu_D > 0$ stating that the average solution of the first algorithm is greater than the average solution of the second algorithm.

6.1. Instance Setting. In this paper, instances generated in the study of Reisi-Nafchi and Moslehi [4] are recognized as the benchmark instances. Different algorithms are evaluated and appraised based on their performance on these benchmark instances. The setting of the benchmark instances is as follows.

Processing times and weights are generated randomly from a uniform distribution within the interval of [1, 10]. When generating due dates, two additional parameters are

(1) **Initialization**
(2) Set $k = 0$, $m = 50$, $c_1 = 0.8$, $c_2 = 0.8$, $w = 0.9$
(3) Initialize $X_i^1, V_i^1, X_i^2, V_i^2$
(4) **Main phase**
(5) **do while** $(k \leq K)$
(6) $k = k + 1$
(7) **for** $i = 1$ **to m do**
(8) $\sigma_i \leftarrow$ Generate sequence $(X_i^1, \ldots, X_m^1, X_i^2, \ldots, X_m^2)$
(9) $\sigma_i \leftarrow$ Local search(σ_i)
(10) Update S^G and S_i^{PB}, $\forall i = 1, \ldots, m$
(11) Update $X_i^1, X_i^2, V_i^1, V_i^2$, $\forall i = 1, \ldots, m$
(12) **end for**
(13) **end do**
(14) **return solution** S^G

Algorithm 2: PSO algorithm.

introduced: tardiness factor τ and the range factor r. Due dates are generated randomly within the range of $[TPT^*(1 - \tau - (r/2)), TPT^*(1 - \tau - (r/2))]$, in which the TPT denotes the sum of processing times from all the jobs. The revenues are generated from a uniform distribution within the interval of $[1, 2 * p_i]$; upper bound Q is generated within the range of $[0, \sum_{i=n_1+1}^{n} w_i]$. The instances are sorted into eight groups according to the parameter setting, the sizes of instances ranging from 20 to 150. In the first four groups, $n_1 = 2n_2$; in the remaining groups, $2n_1 = n_2$. Parameter of τ takes the value of 0.3 in the groups of G01, G02, G05, and G06, and it takes the value of 0.7 in the groups of G03, G04, G07, and G08. The parameter of r takes the value of 0.2 in groups of G01, G03, G05, and G07. In the benchmark instances, there are 1280 instances in total and 20 instances for each parameter setting. The numerical analysis is primarily focused on the average results, which could be obtained from a varied set of results in different parameter settings.

6.2. Numerical Analysis for the LPP Problem. As mentioned above, the performance of PSO and heuristic is compared and weighed against the results in Reisi-Nafchi and Moslehi [4]. The numerical results of GA and *CPLEX* solution in Reisi-Nafchi and Moslehi [4] are obtained from the authors. In small instances, the performance of these algorithms along with the GA algorithm is compared with the *CPLEX* solution, while in large instances, the illustrations include the performance of PSO, heuristic, and GA being compared with each other. It is noteworthy that the CPU time of the heuristic is not shown in the corresponding tables since the proposed heuristic algorithm itself could solve all of the instances within 0.1 seconds.

Table 1 displays the computational results of integer-programming formulation. According to the results, it could be seen that the existing mathematical model could solve problem instances with up to 60 jobs. It could also be seen that the CPU time of G03 and G07 is significantly longer than other groups, when $\tau = 0.3$ and $r = 0.6$. The maximum CPU time in these groups exceeds 1000 seconds, indicating that it is more difficult for the system to search for the optimal solutions in these groups. The results further indicate that the CPU time of the CPLEX solver increases exponentially and thus the larger problem instances cannot be solved using the CPLEX solver. Therefore, heuristic and metaheuristic algorithms are used to solve bigger problem instances.

Table 2 reports the APD values of the three algorithms for the *LPP* problem. As shown in the table, the PSO algorithm performs best in small instances in terms of the solution quality. The solutions of PSO are only 0.01 percent away from the CPLEX solutions. Besides, the maximal APD of PSO is lower than 0.1%, which is lower than other algorithms. On the other hand, though the heuristic deviates furthest from CPLEX solutions, the average APD of the heuristic is lower than 1.5%. This observation also indicates the effectiveness of the heuristic algorithm in solving the small problem instances.

The paired t-test between heuristic and GA found the p value to be 0.00105. Since the p value is less than α (i.e., $0.00105 < 0.05$), the null hypothesis is rejected. The rejection of the null hypothesis state that the average value performance of the GA is statistically better than the performance of the heuristic algorithm at the 95% significance level. Similarly, the paired t-test between heuristic and PSO found the p value to be 0.0024, which indicates the better performance of PSO over heuristic on average solution value. The paired t-test between and GA found the p value to be 0.1955. This analysis indicates that on average results of GA and PSO are not significantly different statistically. However, the average RPD of PSO is slightly better than the average RPD of GA.

Table 3 shows the results of numerical experiments for the *LPP* problem. From Table 3, it could be seen that the PSO also outperforms the algorithms of GA and heuristic for the large instances. The *RPD* of PSO is lower than those of other algorithms with only a few exceptions. The average *RPD* of PSO in all groups is only 0.05, indicating the robustness of PSO for solving the *LPP* problem. Nonetheless, observation from the perspective of CPU time somehow provides a different picture. The average CPU time of PSO is 183.54

Table 1: CPU time of CPLEX solutions in the *LPP* problem.

Group	n	CPU		
		Avg	Min	Max
G01	20	0.09	0.02	0.98
	30	0.37	0.07	3.27
	40	1.5	0.18	7.53
	60	1.51	0.41	15.63
G02	20	0.03	0.02	0.15
	30	0.08	0.06	0.1
	40	0.22	0.17	0.3
	60	0.4	0.31	0.64
G03	20	0.39	0.02	1.18
	30	4.3	0.09	15.07
	40	28.78	0.24	121.75
	60	308.44	0.53	1314.29
G04	20	0.07	0.02	0.64
	30	0.34	0.07	1.46
	40	1	0.17	4.59
	60	11.7	0.48	138.64
G05	20	0.06	0.02	0.4
	30	1.45	0.14	12.28
	40	1.16	0.16	3.85
	60	27.34	0.76	201.91
G06	20	0.04	0.02	0.11
	30	0.4	0.13	1.93
	40	0.22	0.12	0.64
	60	1.77	0.58	8.1
G07	20	0.32	0.08	0.98
	30	4.51	0.55	16.78
	40	35.92	0.48	200.73
	60	519.62	23.01	1442.53
G08	20	0.06	0.02	0.48
	30	0.43	0.08	1.88
	40	0.83	0.17	3.21
	60	7.33	0.59	88.78

seconds, which is longer than the CPU time of GA. The CPU time of both metaheuristics increases significantly as the size of instances increases. Thus, when the size of problem instances becomes larger than the upper bound of 150, the performance of the heuristic algorithm might be more prominent and noteworthy, which is projected through its efficiency. Besides, it is noteworthy to point out that the average value and the maximal value of *RPD* for the heuristic algorithm are 1.21 percent and 4.61 percent, respectively, which also justify the use of heuristic for solving bigger problem instances.

The paired t-test among different algorithms for large problem instances shows a similar indication as to the small problem instances. The PSO and GA are statistically better than the heuristic algorithm. The performances of GA and PSO are not significantly different statistically. However, the average RPD of 0.05 for PSO is better than the average RPD of 0.28 for GA. Based on the discussion above, it could be concluded that both PSO and heuristic could provide reliable solutions. PSO provides better solutions than heuristic and competitive solutions as the GA algorithm, while the heuristic could solve the problem in a shorter CPU time.

6.3. Numerical Analysis for TPP Problem. Table 4 conveys the CPU time for the mathematical model of the *TPP* problem. The *TPP* problem maximizes the net revenue of agent A, subject to an upper bound of the total weighted number of tardy jobs from agent B. As shown in the table, the mathematical model formulation provided in this paper manifests how it could only solve instances with the size limit of 20 jobs, which might be owing to the computational complexity of the problem scenario. In addition to that, the CPU time consumed for the mathematical model is longer than other settings when the tardiness factor takes a value of 0.3 instead of 0.7. In G05, the average CPU time to get the CPLEX solutions is 2089.13 seconds. For one of the instances belonging to this group, it takes more than ten hours (38991.2 seconds) to find the solution.

Table 5 displays the results of PSO and the heuristic algorithm for the small instances. As shown in the table, PSO outperforms the heuristic in all groups in terms of the objective function value. It deviates from the CPLEX solutions only by 0.11 percent. Contrarily, the solution quality of the heuristic algorithm is not as good as PSO; the average of the APD values from the five groups (G03, G04, G06, G07, and G08) exceeds 10 percent. Although it appears that the heuristic in this paper does not perform well for the *TPP* problem, it is noteworthy to mention that the heuristic could still be used as a good comparison for PSO due to its efficiency. The paired t-test between heuristic and PSO for small problem instances found the p value to be 0.00338. The analysis indicates that the average performance of the PSO is statistically better than the average performance of the heuristic algorithm at the 95% significance level.

The numerical results of PSO and heuristic in the *TPP* problem are presented in Table 6 to showcase larger problem instances. The results indicate that the PSO still shows its superiority over the heuristic. It could be concluded that the PSO can provide a better solution for the problem considered in this paper. The heuristic can generate a solution in a shorter time with an acceptable percentage deviation. The paired t-test between heuristic and PSO for large problem instances found the p value to be 2.3×10^{-9} . The analysis strongly suggests that the average performance of the PSO is statistically better than the average performance of the heuristic algorithm at the 95% significance level.

The numerical analysis performed for *LPP* and *TPP* problem indicates the superior performance of PSO over existing algorithms. The superior performance can be attributed to the hybridization of PSO with local search schemes. The superior performance of the hybrid PSO algorithm indicates that the algorithm has the potential to solve different problems arising in different industries. In past, PSO

Table 2: Comparison of algorithms of *LPP* problem in small instances.

Group	n	GA			Heuristic		PSO		
		Average ABS	Average APD	Average CPU time	Average ABS	Average APD	Average ABS	Average APD	Average CPU time
G01	20	2518.30	0	0.94	2517.60	0.03	2518.35	0	0.56
	30	5682.55	0.02	1.95	5680.75	0.05	5683.30	0	1.96
	40	9663.80	0.01	3.9	9658.3	0.06	9664.60	0	5
	60	21319.3	0.01	8.09	21317	0.02	21321.25	0	17.75
G02	20	2460	0.01	1.01	2458.55	0.02	2460.25	0	0.6
	30	5166	0	1.66	5165.15	0	5166.10	0	2.12
	40	9784.05	0	3.54	9784.05	0	9784.40	0	5.38
	60	21093	0	9	21093.15	0	21093.15	0	19.13
G03	20	920.50	0.11	1.94	904.70	2.25	921.30	0.03	0.47
	30	1951.60	0.21	3.76	1940.90	0.7396	1954.40	0.06	1.89
	40	3386.15	0.33	7.01	3370.60	0.81	3394.65	0.05	4.58
	60	7149.85	0.34	15.82	7119.25	0.749	7168.10	0.07	16.25
G04	20	927.95	0.27	1.41	921.25	1.11	929.90	0.01	0.51
	30	2104.65	0.24	3.01	2067.50	2.019	2109.15	0.04	1.79
	40	3606.20	0.26	6.16	3555.45	1.81	3616.10	0.02	4.59
	60	8228.95	0.11	13.44	8108.20	1.56	8236.50	0.02	16.62
G05	20	3945.65	0.04	1.74	3943.60	0.09	3947.25	0	0.79
	30	9043.75	0.02	4.44	9034.70	0.13	9045.20	0	2.90
	40	16682.30	0.03	8.84	16683.30	0.03	16687.75	0	7.73
	60	35684.45	0.03	20.79	35681.90	0.03	35684.45	0	27.86
G06	20	4051.50	0.09	1.67	4035.35	0.44	4052.10	0	1.07
	30	9087.05	0.02	3.36	9045	0.49	9088.50	0	4.19
	40	15829.60	0	6.10	15762.45	0.44	15830.40	0	10.50
	60	36173.25	0	13.8	35997.25	0.5	36174.90	0	37.70
G07	20	1179.8	0.16	1.70	1145.80	3.20	1181.65	0	0.38
	30	2422.85	0.97	3.73	2335.45	4.52	2446.05	0.02	1.41
	40	5065.50	0.69	7.16	5096.70	2.53	4971.40	0.08	3.60
	60	10551.75	1.17	19.06	10276.80	3.77	10674.25	0.02	13.17
G08	20	1414.75	0.02	0.88	1361.90	4.16	1414.90	0	0.40
	30	3572.65	0.04	2.76	3422.10	4.65	3573.50	0	1.54
	40	6018.05	0.13	5.04	5747.70	4.62	6026	0	4.02
	60	13176.25	0.05	11.7	12704.15	3.66	13180.60	0.01	14.58
Average		8745.69	0.17	6.27	8685.52	1.39	8750.01	0.01	7.22

Table 3: Comparison of algorithms for *LPP* problem in large problem instances.

Group	n	GA			Heuristic		PSO		
		Average ABS	Average RPD	Average CPU time	Average ABS	Average RPD	Average ABS	Average RPD	Average CPU time
G01	80	39999.70	0	18.36	39996.55	0.01	40001.35	0	47.17
	100	60348.05	0	33.35	60344.20	0.01	60351	0	98.50
	120	90745.90	0	53.99	90746	0	90747.45	0	175.65
	150	136730.50	0	95.76	136730.25	0	136732.45	0	349.23
G02	80	40321.55	0	19.09	40321.80	0	40322.25	0	50.24
	100	59533.70	0	35.84	59533.25	0	59534.45	0	106.82
	120	88112.60	0	55.2	88112.85	0	88113.20	0	193.73
	150	135944.40	0	93.05	135945.20	0	135945.20	0	375.07
G03	80	13218.35	0.18	33.01	13169.90	0.57	13241.65	0	43.49
	100	21454.10	0.10	51.24	21390.30	0.41	21474.45	0	90.17
	120	29716.55	0.24	75.88	29621.80	0.56	29772.05	0.04	154.81
	150	45783.90	0.17	144.39	45646.40	0.47	45850.05	0.02	333.35

Table 3: Continued.

Group	n	GA			Heuristic		PSO		
		Average ABS	Average RPD	Average CPU time	Average ABS	Average RPD	Average ABS	Average RPD	Average CPU time
G04	80	16005.40	0.08	27.18	15764.05	1.63	16017	0	43.59
	100	23260.65	0.13	43.34	22756	2.39	23290.40	0	86.89
	120	33656.80	0.11	64.35	33128.60	1.77	33688.35	0.02	148.34
	150	55188.40	0.03	113.93	54435.75	1.40	55145.45	0.13	297.54
G05	80	65441.85	0.02	41.28	65444.25	0.02	65454.45	0	72.36
	100	105339.85	0.01	69.83	105337.10	0.01	105351.90	0	152.88
	120	144242	0.01	130.24	144235.25	0.01	144246.15	0	273.69
	150	224901.75	0.01	201.08	224900.70	0.01	224919.10	0	565.53
G06	80	67786.35	0	27.94	67596.85	0.30	67786.15	0	95.73
	100	102016.45	0	58.30	101679.35	0.33	102016.30	0	200.17
	120	142779.90	0	93.90	142188.55	0.41	142761.25	0.02	332.17
	150	229024.20	0	151.58	228191.40	0.36	228882.65	0.06	663.42
G07	80	18552.30	1.46	48.56	18146.20	3.59	18823.95	0	11
	100	31254.30	2.01	94.25	31044.10	2.69	31896.40	0	64.70
	120	44185.65	2.16	154.37	43879.85	2.88	45117.95	0.10	116.44
	150	66529.35	1.99	327.84	66258.80	2.39	67843.05	0.04	240.82
G08	80	23772	0.02	24.75	22833.25	4.07	23770	0.03	36.53
	100	36552.20	0.13	41.10	34912.65	4.67	36548.90	0.10	67.50
	120	55435.10	0	57.59	53400	3.77	55273.10	0.30	123.48
	150	85601.85	0	104.03	84987.10	3.98	82243.20	0.74	262.19
Average		72919.86	0.28	80.77	72583.70	1.21	72911.29	0.05	183.54

Table 4: CPU time of CPLEX solutions in *TPP* problem.

Group	n	CPU		
		Avg	Max	Min
G01	20	202.06	3001.29	0.29
G02	20	7.46	117.49	0.18
G03	20	75.50	428.41	0.20
G04	20	5.19	27.88	0.13
G05	20	2089.13	38991.2	0.19
G06	20	9.69	156.75	0.18
G07	20	75.60	448.80	2.80
G08	20	9.53	49.89	0.71

Table 5: Comparison of algorithms in *TPP* problem in small instances.

Group	n	Heuristic		PSO		
		Average ABS	Average APD	Average ABS	Average CPU time	Average APD
G01	20	116.80	0.68	117.60	0.30	0
G02	20	112.50	2.29	115.15	0.35	0
G03	20	98.30	14.21	112.85	0.53	0.14
G04	20	96.75	11.46	108.50	0.61	0.10
G05	20	110.45	6.02	117.45	0.46	0
G06	20	107.15	10.24	119.50	0.53	0
G07	20	71.30	23.57	92.60	0.35	0.23
G08	20	70.20	29.53	98.15	0.49	0.44
Average		97.93	12.25	110.23	0.45	0.11

Table 6: Comparison of algorithms in *TPP* problem in large instances.

n	Group	Heuristic		PSO		
		Average ABS	Average RPD	Average ABS	Average CPU time	Average RPD
30	G01	173.75	1.96	176.95	0.89	0
40		225.55	3.05	232.60	2.45	0
60		344.05	1.29	348.60	6.35	0
80		463.15	1.10	468.50	17.24	0
100		608.30	1.20	615.50	32.93	0
120		726.05	0.23	727.70	52.49	0
150		892.95	0.23	894.95	105.01	0
30	G02	169.90	0.61	170.85	0.90	0
40		239.55	0.15	239.90	2.20	0
60		363.25	0	363.25	6.94	0
80		485.35	0.09	485.80	17.87	0
100		601.40	0.19	602.60	40.77	0
120		723.30	0.05	723.65	64.81	0
150		903.70	0	903.70	113.56	0
30	G03	146.90	11.37	164.65	1.86	0
40		204.55	9.94	225.35	4.10	0
60		292.95	10.14	323.45	13.87	0
80		414.80	7.53	447.30	36.97	0
100		537.10	7.21	575.75	70.42	0
120		581.60	8.15	625.75	122.50	0
150		807.35	5.61	852.75	247.89	0
30	G04	146.40	14.27	169.85	2.41	0
40		200.45	9.95	221.75	5.64	0
60		308.90	9.59	341.15	18.73	0
80		419.95	8.50	458	53.93	0
100		549.25	8.75	601	114.19	0
120		633.95	10.49	705.30	203.20	0
150		753.05	11.06	843.50	371.27	0
30	G05	170.25	6.08	180.75	1.58	0
40		234.40	3.34	242.40	3.72	0
60		346.85	4.61	363.10	12.29	0
80		463.80	2.58	476.40	28.42	0
100		590.40	2.65	606.35	54.21	0
120		692.50	3.94	720.15	95.59	0.01
150		853	2.75	877.15	205.54	0
30	G06	171.55	7.68	185.90	1.72	0
40		211.50	7.34	227.20	4.01	0
60		343.10	8.22	373.85	13	0
80		431.15	6.29	460.10	26.34	0
100		570.90	5.90	606.85	55.61	0
120		647.70	8.14	704.20	109.97	0
150		822.70	6.22	877.15	198.15	0
30	G07	114.20	23.30	146.70	1.14	0
40		147.10	23.20	190.40	2.75	0
60		226.35	20.75	284.50	9.13	0
80		297.55	21.60	377.20	20.83	0
100		368.55	20.70	462.10	40.88	0
120		460.10	19.70	570	72.39	0
150		559.60	18.74	685.60	142.20	0
30	G08	131.35	23.49	171.15	1.96	0
40		158.10	26.43	214.60	4.81	0
60		243.80	24.31	321.50	18.36	0
80		326.85	23.24	424.55	50.35	0
100		396	24.61	525.35	101.70	0
120		493.05	25.71	662.35	181.65	0
150		588.45	25.46	788.05	374.11	0
Average		428.18	9.64	468.49	63.64	0

has been successfully used to solve the option price model (Sharma et al., [34]). In the future, the hybrid PSO can be applied to solve different pricing and index-based models such coupon bond model (Jin et al. [35]), option pricing model (Jin et al., [36]), and risk index model (Jin and Zhu, [37]).

7. Conclusions

This paper works on the order acceptance two-agent scheduling problems, which are regarded as an NP-hard problem. This paper contemplates and examines the two variants of the problem named the *LPP* problem and the *TPP* problem. The objective function of the *LPP* problem is to maximize the total revenues of accepted orders minus the total weighted lateness of the jobs from agent A. The objective function of the *TPP* problem is to maximize the total revenues of accepted orders minus the total weighted tardiness of jobs from agent A. Both variants of the problem impose a constraint that the sum of the weighted number of tardy jobs belonging to agent B is not allowed to exceed the given upper bound. Algorithms including a metaheuristic (PSO) and a heuristic are proposed to solve these problems. Additionally, a mathematical model is provided. The proposed algorithms are designed in such a way so that it can decode and find a solution to both problems that are being considered in this paper.

The numerical results stipulate that the proposed algorithm is competitive with the existing GA algorithm. The solution produced by the proposed PSO algorithm is within less than 1% of the optimal solution for the small problem instances. The PSO could provide better solutions while the heuristic could provide solutions with an acceptable deviation percentage from the PSO solution in shorter CPU time.

In the future, the mathematical model formulation could be improved and tweaked by removing the redundant constraints and decision variables to solve bigger problem instances. Also, experimental efforts could essentially be performed, focusing on the aspects like reducing CPU time of the proposed PSO algorithm.

Data Availability

Data used in this research work are available from Yuvraj Gajpal (yuvraj.gajpal@umanitoba.ca).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by NSERC, grant 318689.

References

[1] Q. Li, D. Zhang, S. Wang, and I. Kucukkoc, "A dynamic order acceptance and scheduling approach for additive manufacturing on-demand production," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3711–3729, 2019.

[2] Y. Gajpal and H. Li, "Single machine scheduling with two agents for total completion time objectives," in *Applied Operational Research: 8th International Conference, ICAOR 2016*ORLAB Analytics, Rotterdam, The Netherlands, 2016.

[3] H. Li, Y. Gajpal, and C. R. Bector, "Single machine scheduling with two-agent for total weighted completion time objectives," *Applied Soft Computing*, vol. 70, pp. 147–156, 2018.

[4] M. Reisi-Nafchi and G. Moslehi, "A hybrid genetic and linear programming algorithm for two-agent order acceptance and scheduling problem," *Applied Soft Computing*, vol. 33, pp. 37–47, 2015.

[5] A. K. Bhardwaj, Y. Gajpal, C. Surti, and S. S. Gill, "Heart: unrelated parallel machines problem with precedence constraints for task scheduling in cloud computing using heuristic and meta-heuristic algorithms," *Software: Practice and Experience*, vol. 50, no. 12, pp. 2231–2251, 2020.

[6] H. H. Guerrero and G. M. Kern, "How to more effectively accept and refuse orders," *Production and Inventory Management Journal*, vol. 29, no. 4, pp. 59–63, 1988.

[7] B. Pourbabai, "A short term production planning and scheduling model," *Engineering Costs and Production Economics*, vol. 18, no. 2, pp. 159–167, 1989.

[8] S. A. Slotnick and T. E. Morton, "Selecting jobs for a heavily loaded shop with lateness penalties," *Computers & Operations Research*, vol. 23, no. 2, pp. 131–140, 1996.

[9] J. B. Ghosh, "Job selection in a heavily loaded shop," *Computers & Operations Research*, vol. 24, no. 2, pp. 141–145, 1997.

[10] H. F. Lewis and S. A. Slotnick, "Multi-period job selection: planning work loads to maximize profit," *Computers & Operations Research*, vol. 29, no. 8, pp. 1081–1098, 2002.

[11] S. A. Slotnick and T. E. Morton, "Order acceptance with weighted tardiness," *Computers & Operations Research*, vol. 34, no. 10, pp. 3029–3042, 2007.

[12] W. O. Rom and S. A. Slotnick, "Order acceptance using genetic algorithms," *Computers & Operations Research*, vol. 36, no. 6, pp. 1758–1767, 2009.

[13] K. Charnsirisakskul, P. M. Griffin, and P. Keskinocak, "Order selection and scheduling with leadtime flexibility," *IIE Transactions*, vol. 36, no. 7, pp. 697–707, 2004.

[14] C. Oğuz, F. Sibel Salman, and Z. Bilgintürk Yalçin, "Order acceptance and scheduling decisions in make-to-order systems," *International Journal of Production Economics*, vol. 125, no. 1, pp. 200–211, 2010.

[15] X. Zhong, J. Ou, and G. Wang, "Order acceptance and scheduling with machine availability constraints," *European Journal of Operational Research*, vol. 232, no. 3, pp. 435–441, 2014.

[16] Y.-Y. Xiao, R.-Q. Zhang, Q.-H. Zhao, and I. Kaku, "Permutation flow shop scheduling with order acceptance and weighted tardiness," *Applied Mathematics and Computation*, vol. 218, no. 15, pp. 7911–7926, 2012.

[17] D. Lei and X. Guo, "A parallel neighborhood search for order acceptance and scheduling in flow shop environment," *International Journal of Production Economics*, vol. 165, pp. 12–18, 2015.

[18] J. Ou and X. Zhong, "Bicriteria order acceptance and scheduling with consideration of fill rate," *European Journal of Operational Research*, vol. 262, no. 3, pp. 904–907, 2017.

[19] A. Noroozi, M. M. Mazdeh, M. Heydari, and M. Rasti-Barzoki, "Coordinating order acceptance and integrated production-distribution scheduling with batch delivery

considering Third Party Logistics distribution," *Journal of Manufacturing Systems*, vol. 46, pp. 29–45, 2018.

[20] Y. L. T. V. Silva, A. Subramanian, and A. A. Pessoa, "Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times," *Computers & Operations Research*, vol. 90, pp. 142–160, 2018.

[21] H. K. Sarvestani, A. Zadeh, M. Seyfi, and M. Rasti-Barzoki, "Integrated order acceptance and supply chain scheduling problem with supplier selection and due date assignment," *Applied Soft Computing*, vol. 75, pp. 72–83, 2019.

[22] X. Li and J. A. Ventura, "Exact algorithms for a joint order acceptance and scheduling problem," *International Journal of Production Economics*, vol. 223, p. 107516, 2020.

[23] S. Wang and B. Ye, "Exact methods for order acceptance and scheduling on unrelated parallel machines," *Computers & Operations Research*, vol. 104, pp. 159–173, 2019.

[24] G.-H. Wu, C.-Y. Cheng, H.-I. Yang, and C.-T. Chena, "An improved water flow-like algorithm for order acceptance and scheduling with identical parallel machines," *Applied Soft Computing*, vol. 71, pp. 1072–1084, 2018.

[25] V. P. Palakiti, U. Mohan, and V. K. Ganesan, "Order acceptance and scheduling in a parallel machine environment with weighted completion time," *European J. of Industrial Engineering*, vol. 12, no. 4, pp. 535–557, 2018.

[26] S.-H. Chen, Y.-C. Liou, Y.-H. Chen, and K.-C. Wang, "Order acceptance and scheduling problem with carbon emission reduction and electricity tariffs on a single machine," *Sustainability*, vol. 11, no. 19, p. 5432, 2019.

[27] J. Li, X. Zeng, C. Liu, and X. Zhou, "A parallel Lagrange algorithm for order acceptance and scheduling in cluster supply chains," *Knowledge-Based Systems*, vol. 143, pp. 271–283, 2018.

[28] Z. Wang, Y. Qi, H. Cui, and J. Zhang, "A hybrid algorithm for order acceptance and scheduling problem in make-to-stock/make-to-order industries," *Computers & Industrial Engineering*, vol. 127, pp. 841–852, 2019.

[29] M. Yavari, M. Marvi, and A. H. Akbari, "Semi-permutation-based genetic algorithm for order acceptance and scheduling in two-stage assembly problem," *Neural Computing and Applications*, vol. 32, no. 8, pp. 2989–3003, 2020.

[30] H. Li, Y. Gajpal, and C. R. Bector, "A survey of due-date related single-machine with two-agent scheduling problem," *Journal of Industrial & Management Optimization*, vol. 13, no. 5, p. 1, 2019.

[31] P. Melchiors, R. Leus, S. Creemers, and R. Kolisch, "Dynamic order acceptance and capacity planning in a stochastic multi-project environment with a bottleneck resource," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 459–475, 2018.

[32] S. N. Sahu, Y. Gajpal, and S. Debbarma, "Two-agent-based single-machine scheduling with switchover time to minimize total weighted completion time and makespan objectives," *Annals of Operations Research*, vol. 269, no. 1-2, pp. 623–640, 2018.

[33] H. Li, Y. Gajpal, C. Surti, D. Cai, and A. K. Bhardwaj, "Nature-inspired metaheuristics for two-agent scheduling with due date and release time," *Complexity*, vol. 2020, Article ID 1385049, 13 pages, 2020.

[34] B. Sharma, R. K. Thulasiram, and P. Thulasiraman, "Normalized particle swarm optimization for complex chooser option pricing on graphics processing unit," *The Journal of Supercomputing*, vol. 66, no. 1, pp. 170–192, 2013.

[35] T. Jin, Y. Sun, and Y. Zhu, "Time integral about solution of an uncertain fractional order differential equation and application to zero-coupon bond model," *Applied Mathematics and Computation*, vol. 372, p. 124991, 2020.

[36] T. Jin, H. Ding, H. Xia, and J. Bao, "Reliability index and asian barrier option pricing formulas of the uncertain fractional first-hitting time model with caputo type. chaos," *Solitons & Fractals*, vol. 142, Article ID 110409, 2020.

[37] T. Jin and Y. Zhu, "First hitting time about solution for an uncertain fractional differential equation and application to an uncertain risk index model," *Chaos, Solitons & Fractals*, vol. 137, p. 109836, 2020c.

Hindawi
Complexity
Volume 2021, Article ID 6670288, 15 pages
https://doi.org/10.1155/2021/6670288

Research Article

Dynamic Large-Scale Server Scheduling for IVF Queuing Network in Cloud Healthcare System

Yafei Li,[1] Hongfeng Wang,[1] Li Li,[2] and Yaping Fu[3]

[1]*College of Information Science and Engineering, Northeastern University, Shenyang, China*
[2]*Xikang Healthcare Technology Co., Ltd, Shenyang, China*
[3]*School of Business, Qingdao University, Qingdao, China*

Correspondence should be addressed to Hongfeng Wang; hfwang@mail.neu.edu.cn

Received 6 November 2020; Revised 22 December 2020; Accepted 11 January 2021; Published 21 January 2021

Academic Editor: Rui Wang

As one of the most effective medical technologies for the infertile patients, in vitro fertilization (IVF) has been more and more widely developed in recent years. However, prolonged waiting for IVF procedures has become a problem of great concern, since this technology is only mastered by the large general hospitals. To deal with the insufficiency of IVF service capacity, this paper studies an IVF queuing network in an integrated cloud healthcare system, where the two key medical services, that is, egg retrieval and transplantation, are assigned to accomplish in the general hospital, while the routine medical tests are assigned into the community hospital. Based on continuous-time Markov procedure, a dynamic large-scale server scheduling problem in this complicated service network is modeled with consideration of different arrival rates of multiple type of patients and different service capacities of multiple servers that can be defined as doctors of the general hospital. To solve this model, a reinforcement learning (RL) algorithm is proposed, where the reward functions are designed for four conflicting subcosts: setup cost, patient waiting cost, penalty cost for unsatisfied patient personal preferences, and medical cost of patient. The experimental results show that the optimal service rule of each server's queue obtained by the RL method is significantly superior to the traditional service rule.

1. Introduction

According to data released by the Chinese National Health Commission, the infertility rate of couples of child-bearing age in China has risen from 3% to 12%–15% in the past two decades, and infertility has become a younger development trend. It is estimated that about 50 million women are infertile, 66% of whom are below 30 years of age. For giving birth to a baby, most of them try in vitro fertilization (IVF) technology, which places very high requirements for doctors to improve success rate [1]. In China, only doctors in a limited number of general hospitals are qualified for mastering this technology. Due to the limited medical resources of general hospitals, patients have to face a very long wait for IVF procedures. However, the community hospitals are in sharp contrast with the status quo of general hospitals due to backward medical resources and low technical level of medical staff. In very recent years, cloud healthcare has attracted considerable attention of scholars and practitioners, which can fully realize the integration of medical resources of general and community hospitals. Patients can quickly enjoy homogeneous medical services by telemedicine, also reducing the medical costs. Therefore, an integrated problem of meeting the needs of IVF patients and making the medical resources of general hospitals fully utilized is considered in this paper.

Due to the importance of improving the efficiency of system, the resource allocation or scheduling problems in a lot of complex systems, such as manufacturing system [2–4], supply chain system [5], and service system [6–8], had gained great concerns by the researchers from OR community in recent years. It is noticeable that the relevant works on medical resource allocation in healthcare system mostly focused on the scheduling of hospital beds and

operating rooms [9, 10]. Huang et al. [11] studied a patient scheduling problem in EDs, which was modeled as a multiclass queuing network with service deadlines and feedback paths. The proposed scheduling strategy could effectively alleviate the congestion of emergency departments. Erdogan et al. [12] proposed a new random integer programming model to study the dynamic sequencing and scheduling of patients. In this case, the patient's request was not in accordance with the first-come first-service (FCFS) rule, and the scheduler would reserve capacity for emergency patients. He et al. [13] applied hybrid robust random method to describe a scheduling problem of emergency room patients. Through the developed dynamic scheduling algorithm, the matching of doctors and patients was realized. Moreover, researchers have begun to focus on the influence of patient preference and selection behavior upon medical resource allocation recently [14–17]. Dogru et al. [18] developed an appointment scheduling model suitable for primary care settings and provided patients with an ideal appointment schedule based on the patient's order of appointment while meeting patient preferences. Schütz and Kolisch [19] considered a scheduling problem of allocating scarce resources to different customer categories and service types in the case of cancellation, missed appointments, and overbooking. In a patient queuing network with multiple categories, Truong [20] examined a dynamic advance scheduling problem with two patient categories and developed an optimal dynamic scheduling algorithm, which could fully adapt to daily changes in demand and capacity. An unlimited server service system with a limited-service capacity had been considered by Hassin et al. [21], in which servers were sorted by their service rate and arriving clients would join the fastest idle server.

In very recent years, regional medical cooperation has shown great advantages in improving the utilization of medical resources [22]. Especially, the cloud medical system fully realizes the sharing of medical resources among general and community hospitals. Saghafian et al. [23] studied a telemedicine system to decide whether to transfer patients to telemedicine doctors based on the knowledge of community hospital triage nurses. Rajan et al. [24] analyzed a trade-off problem between the treatment speed and quality of chronic patients in the telemedicine system and proved that the benefit-maximizing service rate gradually approaches the social optimal service rate. Erdogan et al. [12] proposed a two-stage stochastic linear model. Taking into account the patient's absentee behavior, they obtained the best planned arrival time for patients in community hospitals using the telemedicine platform and the optimal number of patients that could be arranged for daily telemedicine.

It is noticeable that most solution methods in the abovementioned works focused on the metaheuristic approaches [25–29]. The purpose of this paper is to provide an optimal dynamic resource scheduling rule and service order based on the real-time state of the system instead of the FCFS rule. For this optimal scheduling problem of dynamic tasks, many scholars had adopted reinforcement learning (RL) methods. Huang et al. [30] and Noureddine et al. [31] proposed dynamic resource allocation algorithms based on RL method to optimize resource allocation in real time, respectively. Xiao et al. [32] developed a real-time dynamic task allocation algorithm based on Q-learning. The algorithm was not limited to adapt to its own task arrival process but also fully considered the influence of other agents on the task flow. Asghari et al. [33] proposed a RL-based resource allocation method in order to reduce the cost of system and improve the utilization of resource. Wauters et al. [34] developed a learning-based resource scheduling optimization method to minimize the average delay and total completion time of the project. This paper studies the queuing processes of egg retrieval and transplantation of infertile patients with different arrival rates, and then a dynamic resource allocation algorithm based on reinforcement learning is proposed. Considering the personal preference of patients, the algorithm is able to search for the optimal allocation plan of doctors in general hospitals and the optimal service rules of each doctor's cohort in order to minimize the average total cost of patients.

The rest of this paper is organized as follows. Section 2 presents a detailed description of the IVF process under the telemedicine system. In Section 3, the queuing problem of the IVF in the general hospital is formally defined along with notation, Markov model, objectives, and some preliminary analysis. A resource scheduling algorithm based on reinforcement learning is proposed in Section 4, and the experimental study is presented in Section 5. Finally, concluding remarks are presented in Section 6, followed by some directions for future work.

2. IVF Process in Cloud Healthcare System

The procedure of IVF is a complex and multistage process. In order to improve the utilization rate of resources and service quality, cooperation between general hospitals and community hospitals is necessary. The preexamination process of IVF can be carried out in community hospitals because of its easily learned techniques. Then, through the stimulation of drugs to promote women's ovulation process, once the egg is mature, the doctor will schedule the surgery for her to obtain ovum. After the fertilized egg develops healthily into an embryo, the cultivation stage for embryo is carried out in the community hospital, where the doctors in general hospitals use the telemedicine platform to interpret the embryo report and watch the results of fertilization, division, and blastocyst culture for community hospital doctors. After the embryo sac matures, the patient will return to the general hospital for embryo transfer. Then, they returned to the community hospital for pregnancy examination and they do not need to return to the general hospital because of the remote consultation with doctors in the general hospital through the telemedicine platform. Telemedicine can effectively realize the hierarchical medical services and also can be a great approach to save medical costs and reduce waiting time of IVF patients. Patients in community hospitals can enjoy the excellent services of general hospital just through telemedicine system. We give the IVF flow chart, depending on the actual telemedicine IVF process in the cloud healthcare system (see Figure 1).

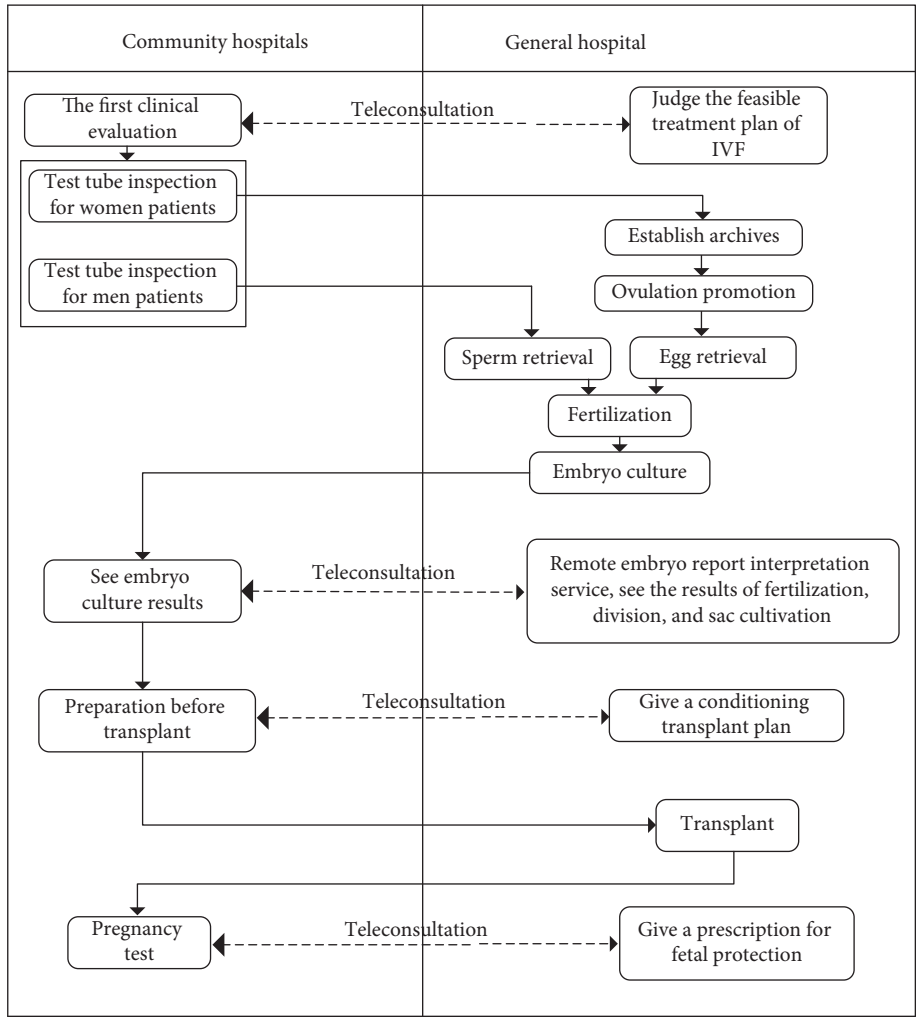


Figure 1: The IVF flowchart in the telemedicine system.

As one of the top-quality scarce resources, doctors in general hospitals have very close relationship with the health present situation. It is obvious from Figure 1 that doctors in general hospitals are the bottleneck that severely restricts the cloud healthcare system for operation, and they only play important roles in the egg retrieval process and the transplant process. Therefore, we just take these two queuing processes into account. The feature of the IVF procedure is time-consuming, so, in the proposed short-term resource scheduling problem, egg retrieval and transplantation are regarded as two different types of patients, and each has its own update arrival process as well as the general service time distribution. After completing the corresponding IVF process in community hospitals, patients line up to enter the queue of general hospitals for egg retrieval or transplant surgery.

3. Dynamic Resource Scheduling Model of IVF

The main problems we tackle in this paper are how to allocate the two types of infertility patients to the service queues of different doctors according to the patients' choice preferences and how to determine the better service rules of each queue. The current optimization and scheduling practice has some obvious shortcomings. First, the service order is based on the rule of FCFS. Second, when considering multiple servers, patients' choice preferences are often neglected or taken into account exactly. Therefore, we formulate a Markov model to improve the current scheduling process addressing each shortcoming. We can fill the gap in the exciting literature on scheduling problem of IVF queuing network.

In this section, we simplify the queuing process of infertile patients in general hospitals to the queuing network shown in Figure 2 according to the IVF process of the cloud medical system in Figure 1. In the cloud medical system, egg retrieval patients and transplant patients may be from different community hospitals, and they share the resource pool of general hospitals.

The notations are given in Table 1.

We can get that the arrival rate of type 1 patients and type 2 patients in general hospital are, respectively, $\lambda_1(t) = \sum_{h=1}^{H} \lambda_{h,1}(t)$ and $\lambda_2(t) = \sum_{h=1}^{H} \lambda_{h,2}(t)$. The service rates of the system serving type 1 and type 2 patients are

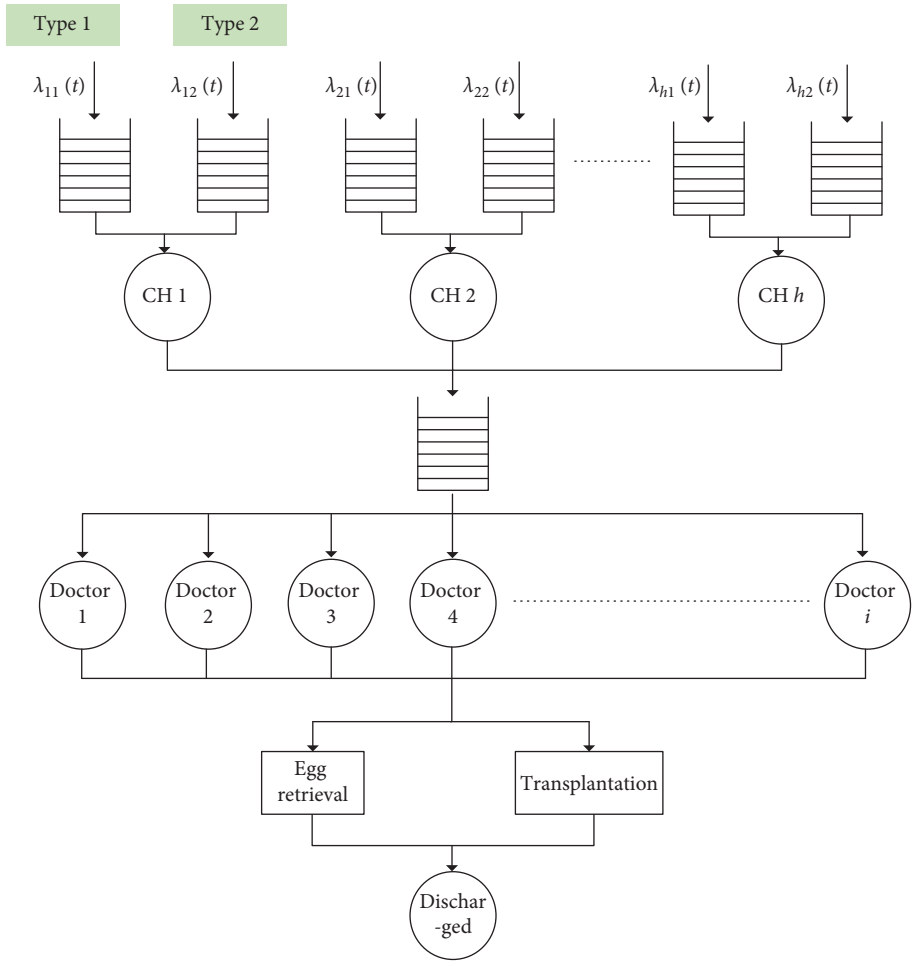


Figure 2: The queuing network of infertile patients in general hospital.

Table 1: Notation.

Sets:	
I	A set of specialists in general hospital
J	A set of types of infertility patients
H	A set of community hospitals
T	A set of time periods in the planning cycle
Indices:	
i	Index of specialists in general hospital
j	Type index of infertile patients
h	Index of community hospitals
t	Index of time periods in the planning cycle
$d(i)$	Index of patients served by doctor i
Parameters:	
$\mathscr{E}_1(t)$	The waiting time cost of the patients in the general hospital during period t
$\mathscr{E}_2$	The setup cost between disease categories 1 and 2 for specialists
$\mathscr{E}_3$	Penalty costs of unmet infertility patients' personal preferences
c_i	Unit medical cost of doctor i
π	Service rules for each doctor queue
$\lambda_{h,j}(t)$	The arrival rate of type j patient in community hospital h at the beginning of the t^{th} time period
$\mu_{i,j}$	The service rate of type i doctor serving type j patient in the general hospital
Random variables:	
$\mathscr{J}_i(t)$	Number of service settings of type i doctor in the t^{th} time period
Decision variables:	
$z_{d(i)}(t)$	In the t^{th} time period, 0 if the personal preference of patient $d(i)$ served by doctor i is met and 1 otherwise
$\mathscr{Y}_{d(i)}(t)$	In the t^{th} time period, 1 if doctor i is assigned to patient $d(i)$ and 0 otherwise

$\mu_1 = \sum_{i=1}^{I} \mu_{i,1}$ and $\mu_2 = \sum_{i=1}^{I} \mu_{i,2}$, respectively. The arrival rate and the service rate are given based on a real data from the actual cloud healthcare system. We model the queuing process of infertile patients in general hospital as the Markov decision process. We presented the state transition diagrams of the two types of patients at different time periods as Figure 3.

In Figure 3, the probability that the number of transplanted patients varies from $n+1$ to n $(0 \le n \le N)$ is $\sum_{i=1}^{I} y_{i2}\mu_2$ in the fixed number of egg retrieval patients, because, in different planned time periods, the service queue of each doctor may contain two types of patients or only one type of patients, or even the queue is empty. In this case, the service rate of the system is changed at any moment depending on the number of doctors serving transplant patients. At different moments, the value of $\sum_{i=1}^{I} y_{i2}\mu_2$ is different. For instance, in the t^{th} time period, the general hospital has a total of I available doctors resources, but only a cohort of 4 doctors have 2 types of patients. Then, given the number of patients in one type, the probability that the total number of patients varies from $n+1$ to n is $4\mu_2$. In the $t+1^{\text{th}}$ time period, suppose that there are 2 types of patients in the service queue of 3 doctors; then, given that the number of patients in one type is m, the probability that the number of patients in type 2 varies from $n+1$ to n is $3\mu_2$.

In the steady state, we give the following balance equation:

For $m = 0$ and $n = 0$,

$$(\lambda_1(t) + \lambda_2(t))P(0,0) = \mu_1 P(1,0) + \mu_2 P(0,1). \tag{1}$$

For $m = 0$ and $n = 1, 2, \ldots, N-1$,

$$\begin{aligned}\left(\lambda_1(t) + \lambda_2(t) + \sum_{i=1}^{I} y_{i2}\mu_2\right)P(0,n) &= \lambda_2(t)P(1,n-1) + \sum_{i=1}^{I} y_{i1}\mu_1 P(1,n) \\ &+ \sum_{i=1}^{I} y_{i2}\mu_2 P(0,n+1).\end{aligned} \tag{2}$$

For $m = 0$ and $n = N$,

$$(\lambda_1(t) + \sum_{i=1}^{I} y_{i2}\mu_2)P(0,N) = \lambda_2(t)P(0,N-1) + \sum_{i=1}^{I} y_{i1}\mu_1 P(1,N). \tag{3}$$

For $m = 1, 2, \ldots, M-1$ and $n = 0$,

$$\begin{aligned}\left(\lambda_1(t) + \lambda_2(t) + \sum_{i=1}^{I} y_{i1}\mu_1\right)P(m,0) &= \lambda_1(t)P(m-1,0) + \sum_{i=1}^{I} y_{i1}\mu_1 P(m+1,0) \\ &+ \sum_{i=1}^{I} y_{i2}\mu_2 P(m,1).\end{aligned} \tag{4}$$

For $m = 1, 2, \ldots, M-1$ and $n = 1, 2, \ldots, N-1$,

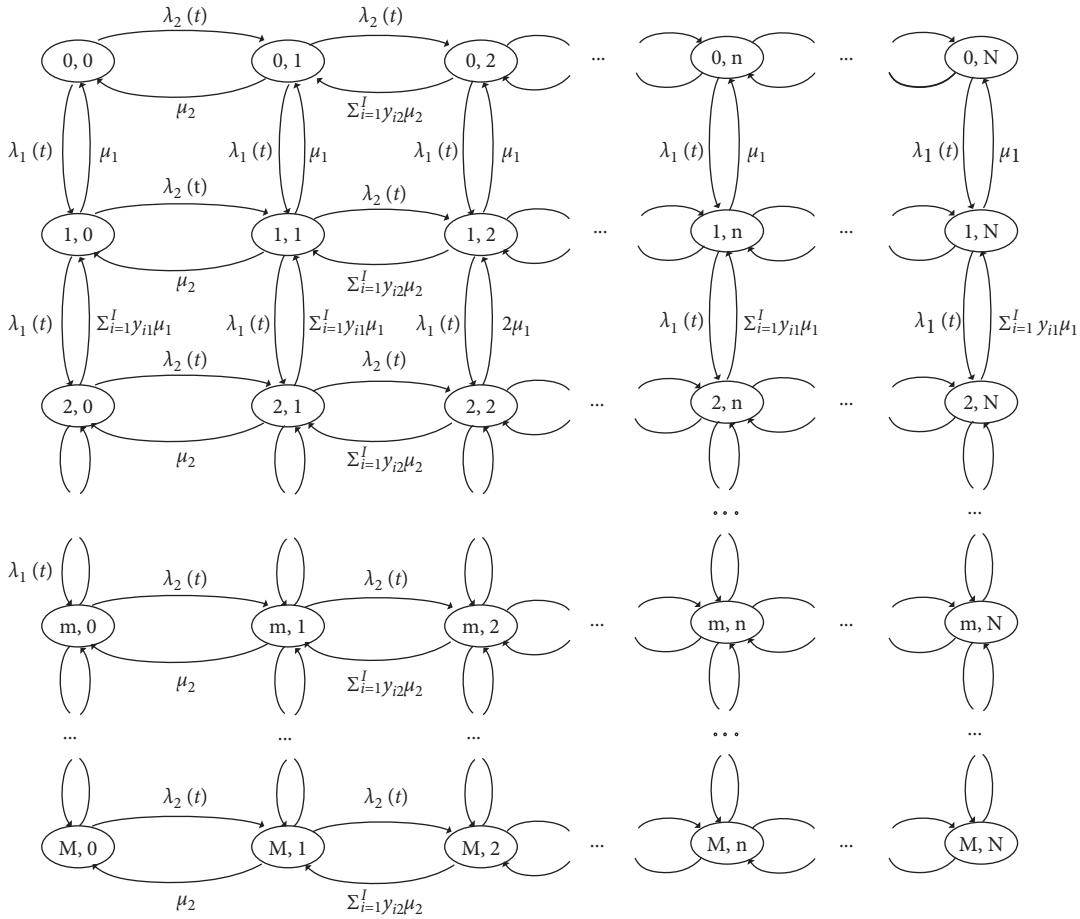


Figure 3: The state transition diagrams of the two types of patients.

$$\left(\lambda_1(t)+\lambda_2(t)+\sum_{i=1}^{I} y_{i1}\mu_1+\sum_{i=1}^{I} y_{i2}\mu_2\right)P(m,n)=\lambda_1(t)P(m-1,n)+\lambda_2(t)P(m,n-1)+\sum_{i=1}^{I} y_{i1}\mu_1 P(m+1,n)+\sum_{i=1}^{I} y_{i2}\mu_2 P(m,n+1). \tag{5}$$

For $m = 1, 2, \ldots, M-1$ and $n = N$,

$$\left(\lambda_1(t)+\sum_{i=1}^{I} y_{i1}\mu_1+\sum_{i=1}^{I} y_{i2}\mu_2\right)P(m,N)=\lambda_1(t)P(m-1,N)+\lambda_2(t)P(m,N-1)+\sum_{i=1}^{I} y_{i1}\mu_1 P(m+1,N). \tag{6}$$

For $m = M$ and $n = 0$,

$$\left(\lambda_2(t)+\sum_{i=1}^{I} y_{i1}\mu_1\right)P(M,0)=\lambda_1(t)P(M-1,0)+\sum_{i=1}^{I} y_{i2}\mu_2 P(M,1). \tag{7}$$

For $m = M$ and $n = 1, 2, \ldots, N-1$,

$$\left(\lambda_2(t) + \sum_{i=1}^{I} y_{i1}\mu_1 + \sum_{i=1}^{I} y_{i2}\mu_2\right)P(M,n) = \lambda_1(t)P(M-1,n) + \lambda_2(t)P(M,n-1) + \sum_{i=1}^{I} y_{i2}\mu_2 P(M,n+1). \tag{8}$$

For $m = M$ and $n = N$,

$$\left(\sum_{i=1}^{I} y_{i1}\mu_1 + \sum_{i=1}^{I} y_{i2}\mu_2\right)P(M,N) = \lambda_1(t)P(M-1,N) + \lambda_2(t)P(M,N-1). \tag{9}$$

The infertile patients in the cloud healthcare system come from different community hospitals; in that sense, the scale of this problem is a large one. In order to effectively solve the proposed problem, the state transition diagram in Figure 3 is simplified. Given the number of patients in type 2, the state transition diagram of patients in type 1 in the t^{th} time period is shown in Figure 4.

We assume that the upper limit of the number of patients in type 1 (i.e., egg retrieval patients) is M. Let the steady-state probability of the number (of m) of egg retrieval patients in the system be $P(Q_j = m)$. The state balance equation for the queuing process of egg retrieval patients is shown below.

For $m = 0$,

$$\lambda_1(t)P(Q_1 = 0) = \sum_{i=1}^{I} y_{i1}\mu_1 P(Q_1 = 1). \tag{10}$$

For $m = 1, 2, \ldots, M-1$,

$$\left(\lambda_1(t) + \sum_{i=1}^{I} y_{i1}\mu_1\right)P(Q_1 = m) = \lambda_1(t)P(Q_1 = m-1) + \sum_{i=1}^{I} y_{i1}\mu_1 P(Q_1 = m+1). \tag{11}$$

For $m = M$,

$$\sum_{i=1}^{I} y_{i1}\mu_1 P(Q_1 = M) = \lambda_1(t)P(Q_1 = M-1). \tag{12}$$

Define the service intensity for egg retrieval patients in the system as $\rho_1 = (\lambda_1(t)/\sum_{i=1}^{I} y_{i1}\mu_1)$; when $m = 0$, we can have

$$P(Q_1 = 0) = \frac{1}{\sum_{m=0}^{\left(\Sigma_{i=1}^{I} y_{i1}-1\right)} (1/m!)\left(\lambda_1(t)/\mu_1\right)^m + \left(1/\left(\sum_{i=1}^{I} y_{i1}\right)!\left(1-\rho_1\right)\right)\left(\lambda_1(t)/\mu_1\right)^{\left(\Sigma_{i=1}^{I} y_{i1}\right)}}. \tag{13}$$

When $m \geq 1$,

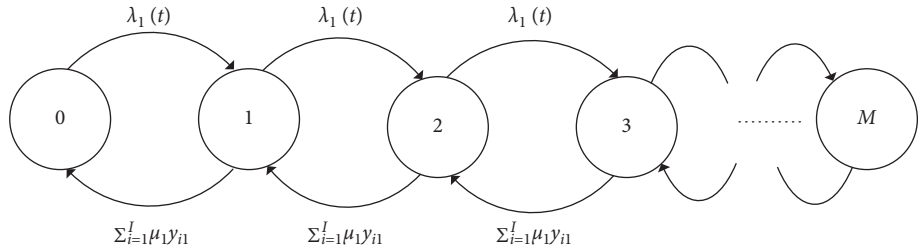


Figure 4: The state transition diagram for type 1 patients.

$$P(Q_1 = m) = \begin{cases} \dfrac{1}{\left(\sum_{i=1}^{I} y_{i1}\right)!}\left(\dfrac{\lambda_1(t)}{\mu_1}\right)^m P(Q_1 = 0), & n \le \sum_{i=1}^{I} y_{i1}, \\ \dfrac{1}{\left(\sum_{i=1}^{I} y_{i1}\right)^{\left(m - \sum_{i=1}^{I} y_{i1}\right)}\left(\sum_{i=1}^{I} y_{i1}\right)!}\left(\dfrac{\lambda_1(t)}{\mu_1}\right)^m P(Q_1 = 0), & n > \sum_{i=1}^{I} y_{i1}. \end{cases} \tag{14}$$

In the steady state, from the state balance equation of the egg retrieval patients, the state transition probability matrix of the egg retrieval patients can be obtained:

$$\begin{bmatrix} & 1-\lambda_1(t) & \lambda_1(t) & \cdots & 0 \\ \sum_{i=1}^{I} y_{i1}\mu_1 & & 1-\left(\sum_{i=1}^{I} y_{i1}\mu_1 + \lambda_1(t)\right) & \lambda_1(t) & \cdots & 0 \\ 0 & \sum_{i=1}^{I} y_{i1}\mu_1 & 1-\left(\sum_{i=1}^{I} y_{i1}\mu_1 + \lambda_1(t)\right) & \lambda_1(t) & \cdots & 0 \\ & \vdots & \cdots & \ddots & \vdots & \\ 0 & \cdots & & \sum_{i=1}^{I} y_{i1}\mu_1 & 1-\sum_{i=1}^{I} y_{i1}\mu_1 & \end{bmatrix}. \tag{15}$$

According to Little's formula, the queue length of egg retrieval patients in the t^{th} time period is

$$L_1(t) = \frac{(\lambda_1(t)/\mu_1)^{\sum_{i=1}^{I} y_{i1}}\rho_1 P(Q_1 = 0)}{\left(\sum_{i=1}^{I} y_{i1}\right)!(1-\rho_1)^2}. \tag{16}$$

The expected number of the egg retrieval patients in the t^{th} time period is

$$L_s(t) = \sum_{m=1}^{M} mP(Q_1 = m). \tag{17}$$

The total service intensity of the system is denoted as $\rho = (\lambda_1(t) + \lambda_2(t)/\sum_{i=1}^{I}\sum_{j=1}^{J} y_{ij}\mu_j)$. The average waiting time for infertile patients in the t^{th} time period is

$$W_1(t) = \frac{L_1(t)}{\lambda_1(t)} = \frac{(\lambda_1(t)/\mu_1)^{\sum_{i=1}^{I} y_{i1}}\rho\rho_1(Q_1 = 0)}{\left(\sum_{i=1}^{I} y_{i1}\right)!(1-\rho_1)^2\lambda_1(t)}. \tag{18}$$

The number of patients served in the t^{th} time period is

$$\sum_{i=1}^{I} D_{i1}(t) = \lambda_1(t) * t - L_s(t). \tag{19}$$

The average number of customers of infertile patients in the t^{th} time period is

$$L_1'(t) = W_1(t)\lambda_1(t) = \frac{\lambda_1(t)}{\mu_1} + \frac{(\lambda_1(t)/\mu_1)^{\sum_{i=1}^{I} y_{i1}}\rho_1 P(Q_1 = 0)}{\left(\sum_{i=1}^{I} y_{i1}\right)!(1-\rho_1)^2}. \tag{20}$$

Furthermore, assume that the upper limit of the number of patients in type 2 (i.e., transplant patients) is N. Let the steady-state probability of the number (of n) of transplant patients in the system be $P(Q_j = n)$. The simplified state transition diagram of transplant patients is shown in Figure 5.

In the queuing system for infertile patients, the service intensity of transplant patients is defined as $\rho_2 = (\lambda_2(t)/\sum_{i=1}^{I} y_{i2}\mu_2)$. Given the number of egg retrieval patients, we can get the number of transplant patients in the t^{th} time period. Hence, we can obtain

$$P(Q_2 = n|Q_1 = m) = \begin{cases} \dfrac{1}{\left(\sum_{i=1}^{I} y_{i2}\right)!}\left(\dfrac{\lambda_2(t)}{\mu_2}\right)^m P(Q_2 = 0|Q_1 = m), & m = 0, 1, \ldots, \sum_{i=1}^{I} y_{i1}, \\ \dfrac{1}{\left(\sum_{i=1}^{I} y_{i2}\right)^{\left(m - \sum_{i=1}^{I} y_{i2}\right)}\left(\sum_{i=1}^{I} y_{i2}\right)!}\left(\dfrac{\lambda_2(t)}{\mu_2}\right)^n P(Q_2 = 0|Q_1 = m), & m = \sum_{i=1}^{I} y_{i1}, \ldots, M, \end{cases} \tag{21}$$

$$P(Q_2 = 0|Q_1 = m) = \left\{ \frac{1}{\sum_{n=0}^{\left(\sum_{i=1}^{I} y_{i2} - 1\right)} (1/n!)\left(\lambda_2(t)/\mu_2\right)^m + \left(1/\left(\sum_{i=1}^{I} y_{i2} - 1\right)!(1 - \rho_2)\right)\left(\lambda_2(t)/\mu_2\right)^{\left(\sum_{i=1}^{I} y_{i2}\right)}} \right.$$

From the conditional probability formula, we can deduce the steady-state probability of the number of transplant patients in the system as

$$P(Q_2 = n) = \sum_{m=0}^{M} P(Q_2 = n|Q_1 = m)P(Q_1 = m), \quad \text{for } n = 0, 1, \ldots, N. \tag{22}$$

When the number of egg retrieval patients is given, the state transition probability matrix of transplant patients can be obtained as

$$\begin{bmatrix} & 1 - \lambda_2(t) & \lambda_2(t) & \cdots & 0 \\ \sum_{i=1}^{I} y_{i2}\mu_2 & & 1 - \left(\sum_{i=1}^{I} y_{i2}\mu_2 + \lambda_2(t)\right) & \lambda_2(t) & \cdots & 0 \\ 0 & \sum_{i=1}^{I} y_{i2}\mu_2 & 1 - \left(\sum_{i=1}^{I} y_{i2}\mu_2 + \lambda_2(t)\right) & \lambda_2(t) & \cdots & 0 \\ & \vdots & \cdots & \ddots & \vdots & \\ 0 & \cdots & & \sum_{i=1}^{I} y_{i2}\mu_2 & 1 - \sum_{i=1}^{I} y_{i2}\mu_2 & \end{bmatrix}. \tag{23}$$

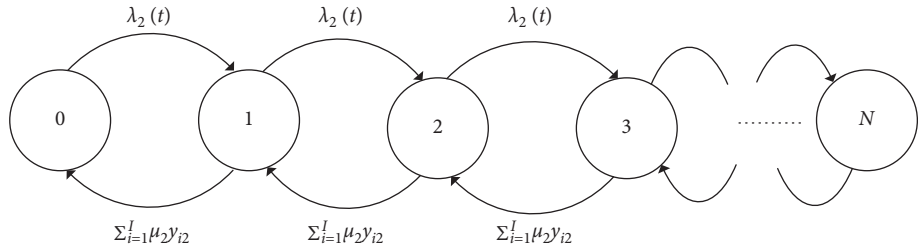


Figure 5: The state transition diagram for type 2 patients.

The queue length of transplant patients in the t^{th} time period is

$$L_2(t) = \frac{(\lambda_2(t)/\mu_2)^{\sum_{i=1}^{I} y_{i2}} \rho_2 P(Q_2 = 0|Q_1 = m)}{\left(\sum_{i=1}^{I} y_{i2}\right)!(1-\rho_2)^2}. \quad (24)$$

The expected number of transplant patients in the t^{th} time period is

$$L_s'(t) = \sum_{n=1}^{N} nP(Q_2 = n|Q_1 = m). \quad (25)$$

The average waiting time for transplant patients in the t^{th} time period is

$$W_2(t) = \frac{L_2(t)}{\lambda_2(t)} = \frac{(\lambda_2(t)/\mu_2)^{\sum_{i=1}^{I} y_{i2}} \rho_2 P(Q_2 = 0|Q_1 = m)}{\left(\sum_{i=1}^{I} y_{i2}\right)!(1-\rho_2)^2 \lambda_2(t)}. \quad (26)$$

The number of transplant patients served in the t^{th} time period is

$$\sum_{i=1}^{I} D_{i2}(t) = \lambda_2(t) * t - L_s'(t). \quad (27)$$

The average number of transplant patients in the t^{th} time period is

$$L_2'(t) = W_2(t)\lambda_2(t) = \frac{\lambda_2(t)}{\mu_2} + \frac{(\lambda_2(t)/\mu_2)^{\sum_{i=1}^{I} y_{i2}} \rho_2 P(Q_2 = 0|Q_1 = m)}{\left(\sum_{i=1}^{I} y_{i2}\right)!(1-\rho_2)^2}. \quad (28)$$

We consider the setup cost when the doctor switches from the current type of patients to another; the aim is to gain the largest equilibrium benefits between community hospitals and general hospital. Define the setting times of each doctor in t time period as $\mathcal{J}_i(t) = \sum_{d(i)}^{D(i)} x_{d,d-1}^{i}$. Let $\mathcal{D}_i(t) = \sum_{j=1}^{J} \mathcal{D}_{ij}(t)$ be the total number of two types of patients served by doctor i in the time period t. The average waiting time and queue length of the two types of infertility patients in the t^{th} time period are, respectively, $W(t) = W_1(t) + W_2(t)$ and $L(t) = L_1(t) + L_2(t)$. In a planning period T, the matching problem between doctors and patients can be well solved according to the model established by us. The objective function is established as follows:

$$\mathcal{F}^{\pi}(t) = \min_{t \in T}\left(\frac{1}{t}\left[\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{d(i)}^{D(i)}\left(\frac{1}{\mu_{i,j}} c_i \mathcal{D}_i(t)\mathcal{Y}_{d(i)}(t) + z_d^i(t)\mathcal{E}_3\right) + \sum_{t=1}^{T}\mathcal{E}_1(t)W(t)L(t) + \sum_{t=1}^{T}\sum_{i=1}^{I}\mathcal{J}_i(t)\mathcal{E}_2\right]\right). \quad (29)$$

The objective function minimizes the total costs, the first term of the objective function represents the medical cost of infertile patients, the second term is the penalty cost of unmet infertility patients' personal preferences, the third term represents the waiting time cost of egg retrieval patients and transplant patients in the system, and the last term is doctors' setup cost. In general, the above objective reflects the interests of both general hospitals and infertile patients in the cloud medical system. In order to avoid a patient's too long waiting time in the system, a four-cost reward function based on reinforcement learning is designed.

4. *Q*-Learning-Based Solution Method

Existing scheduling rules seldom solve the problem of doctor resource scheduling with different service rates in cloud

medical systems. The doctor scheduling that frequently appeared in the existing related literature is based on some given sequence of the patients. In a dynamic environment, doctors' service rate and patients' waiting time are considered as important indices of medical system, especially in cloud medical system with multiple hospitals cooperating with each other. The operating efficiency and the operating costs of the scheduling results greatly varied with the different scheduling rules. Therefore, our purpose is to design an optimal scheduling rule and patient service order for each doctor queue by using reinforcement learning approach. In this section, three-stage dynamic scheduling problems are proposed. At the first step, we divide the planning cycle into hourly dynamic scheduling problems; by different arrival rate and different service rate, we need to decide which patients are assigned to one of the doctors. At the second step, the service rule is presented based on the learning strategies we design. At the third step, we need to make decisions about the number of doctors serving different types of patients from community hospitals in different time periods.

Reinforcement learning is an online actor critic method in machine learning (Sutton et al. [35] and Gao et al. [36]), which obtains certain rewards through interaction with the environment and ultimately maximizes long-term returns. The typical learning algorithm is to update the current state-action pair based on the observed reward and the next state-action pair. Combined with the research questions in this article, we give the algorithm framework based on reinforcement learning in Algorithm 1.

In the IVF queuing system based on the cloud medical system studied in this paper, we define that the state space of the system is composed of the number of patients of the two types at different times and the busyness of the respective queues of doctors in general hospitals. Egg retrieval and transplant patients can only make appointments for related operations within the given appointment time period. We give the system the number of appointments allowed on the day and the strict upper limit of doctor resources. Infertility is a special disease, so the operation must be completed in a given time, even if overtime doctors also have to complete all operations. The average cost of IVF operation much surpasses common operation. Therefore, choosing a method is a key step in cost-saving and drives higher operational efficiency.

Reinforcement learning is an effective dynamic programming method to solve dynamic scheduling problems, and its basic idea is shown in Algorithm 1. It can continuously train based on data to obtain accurate responses to the environment. The main core of reinforcement learning is the design of the reward functions which are guided by the learning system targeted at minimizing the average total cost of the system. However, our total cost consists of four parts: the average service cost, the average waiting time cost, the average setup cost, and the average penalty cost for unmet infertility patients. In order to maximize the long-term total revenue, according to the greedy strategy, we set the reward functions in different states for the four subcosts. After each step is executed, the cumulative reward score of each subcost is treated as the total reward score obtained after the current action is executed, which ultimately maximizes the cumulative reward. The proposed reward functions balance the interests of both doctors and patients. In different planning time periods, different reward and punishment strategies are set according to the state of the system. According to the Markov decision model established in the previous section, the possible state of the system at each moment can be obtained, and, by using the trial-and-error method, we explore all the possible behaviors generated by the current state to find the current maximum return: $\max_{a'} Q(s', a')$.

5. Experimental Study

In this section, we discuss our findings. In particular, we describe the effectiveness of the proposed learning algorithm for reducing the waiting time and the medical cost. The scheduling rules and matching results of patients and doctors can simplify the complex scheduling problem. The IVF queuing network we studied includes one general hospital and 30 community hospitals. The problem raised is to serve hundreds of patients of two different types on six servers (general hospital doctors) with different service rates. As discussed earlier, the patients demand scenarios derived from actual demand for the sake of ensuring the veracity and reliability of this experiment.

This paper studies a resource scheduling problem within a day, where the allowable appointment time period is from 8 : 00 to 12 : 00. We divide the one-day scheduling problem into hourly subscheduling problems. We apply the sequence-based scheduling method and the learning algorithm to compare the efficiency and medical costs. Figure 6 shows a scheduling Gantt chart based on appointment orders (FCFS) of two types of patients in six parallel service desks. If the service desk is free, the patients will automatically join the queue. We are interested in how much better the overall results can be when patients' choice preference is not completely considered. In this experiment, idle servers provide services to patients in the order of appointment. However, these patients with personal preferences have to wait until the doctors of their choice are available. Finally, we calculate that the objective function value under this rule is 50816 yuan.

In Figure 7, we conduct 50 rounds of learning based on the Q-learning algorithm, where the abscissa represents the number of learning rounds and the ordinate represents the total reward value corresponding to each round. Obviously, the learning result curve of the first 40 times is very volatile, because Q-learning algorithm tries to find a better result to balance the four subgoals. The curve finally converges to near 45500, which shows that the results of Q-learning algorithm are better than those of FCFS. Figure 8 is the scheduling Gantt chart given by the 50^{th} round of learning results, which not only shows the doctors and patients match relations but also reflects the patients' personal choice behavior. In order to reduce the long waiting time of patients, the service sequence of patients who arrive at the current time is arranged downstream of the patients who have made

Letter explanation: $Q(s,a)$: learning table about s and a;
s: the state after each step is executed
a: predict the action to be performed next
Require: initialize $Q(s,a)$ arbitrary value;
Repeat (for each episode):
Initialize s
Repeat (for each step of episode):
$A \longleftarrow$ action given by strategy π for s in the learning table $Q(s,a)$.
Take action A, observe the next state s$'$ and reward r
$Q(s,a) \longleftarrow Q(s,a) + \alpha[r + \gamma^* \max_{a'} Q(s', a') - Q(s,a)]$
$s \longleftarrow s'$; $a \longleftarrow a'$;
Until s is terminal

Algorithm 1: Algorithm framework based on reinforcement learning.

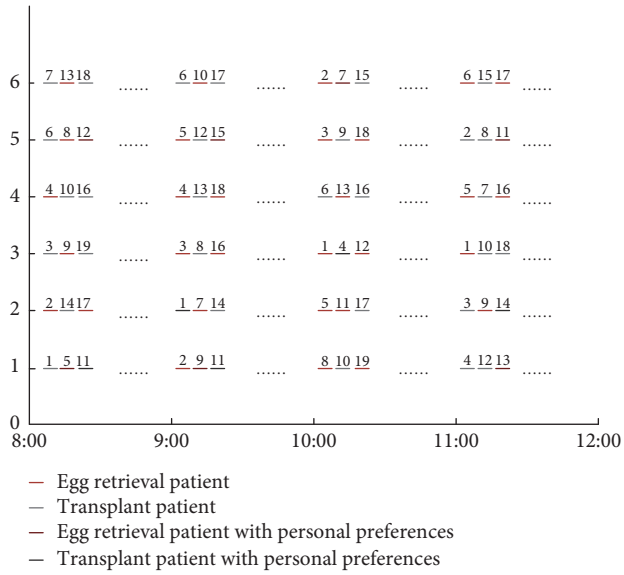


Figure 6: Scheduling Gantt diagram based on sequence rules.

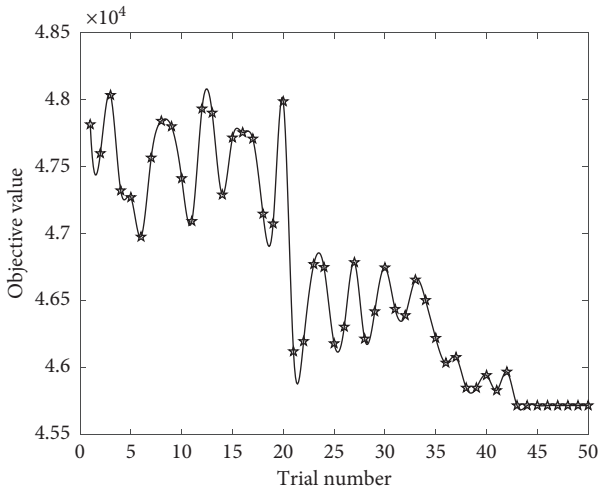


Figure 7: Performance curve of Q-learning scheduling approach.

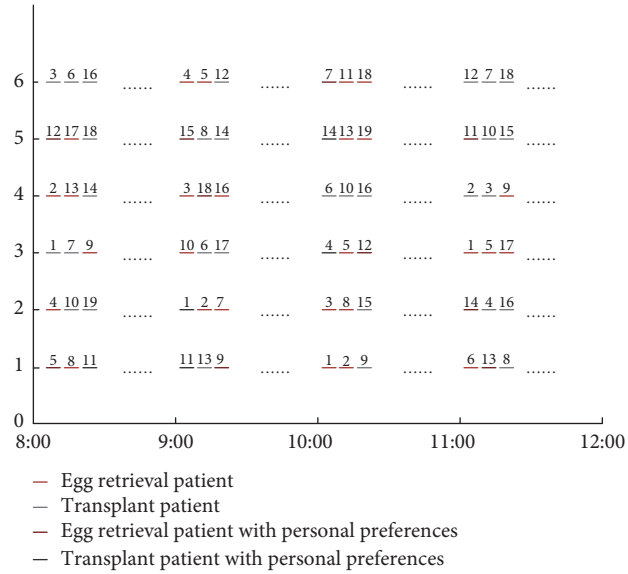


Figure 8: Scheduling Gantt diagram based on Q-learning.

an appointment in the previous time period. For example, for patients who make an appointment at 9 : 00–10 : 00, the Q-learning algorithm determines their service order and which doctor they should be served by, but their service order is scheduled after the patients who make an appointment at 8 : 00–9:00.

The digits in Figure 8 are labeled according to the order of arrival of the patients. Due to the characteristic of large scale, we only give some scheduling results in each time period. Interestingly, the patient service order based on reinforcement learning is almost completely different from that in the FCFS rules, and this is because of the fact that the reinforcement learning algorithm can achieve a better trade-off between the waiting time of patients and the setup cost of doctors, and it also takes into account the service rate of doctors, the different medical costs from doctors, and the personal choice preference of some patients.

Our experimental results can provide decision support for managers. In a queuing system with multiple types of patients and multiple service desks, we can reasonably arrange the number of patients and service order of each service desk according to the choice preference of patients. This has certain reference significance for other organizations with scarce resources. According to experiments conducted on the computer using Python, we can obtain satisfactory results within 3 minutes through the Q-learning. Due to the large scale of the proposed problem, the relatively optimal results that can be obtained within a few minutes are enough to prove that our proposed method has high effectiveness for solving the dynamic scheduling problem with multiple types of patients and multiple service queues.

6. Conclusions and Future Research

In this paper, a dynamic server scheduling problem in a special in vitro fertilization (IVF) queuing network, which is developed in an integrated cloud healthcare system, is investigated in order to address the prolonged waiting problem for IVF service. Based on continuous-time Markov procedure, a mathematical model is established, in which multiple types of patient selection preferences and multiple doctors with different service rates are considered simultaneously. To solve this model, a Q-learning-based solution method is proposed, where the reward functions are designed according to four conflicting cost functions: setup cost, waiting cost, penalty cost, and medical cost. A series of simulation experiments that are generated according to the actual data from Shenyang cloud hospital are carried out to validate the performance of the proposed reinforcement learning (RL) method for the investigated dynamic server scheduling problem in IVF queuing network.

The main contributions in this work lie in three aspects. Firstly, the IVF queueing network developed in cloud healthcare system is helpful to cope with the prolonged waiting problem of IVF medical service. Experimental results show that the waiting costs of patients decrease significantly in this integrated IVF queueing network. Secondly, dynamic server scheduling decision, that is, allocation of doctors in general hospital, is important to improve the efficiency of IVF medical service system. Our developed Markov model can exhibit a very nice flexibility in terms of both patient selection preference and medical resource utilization. Finally, the RL method is effective in solving the investigated problem in this paper. Based on the experimental results, our proposed Q-learning-based solution method significantly outperforms the traditional service rule. In general, the proposed methodology in this paper can not only make good use of the bottleneck medical resources in general hospital but also improve the utilization rate of idle medical resources in community hospital and it enables providing management insights for the cooperation of hospitals in hierarchical medical system.

This work can be extended in future researches. Firstly, we can consider the occupation of resources in the

telemedicine service process of IVF in community hospitals, as well as the influence of the success rate of surgery on the queuing system. Secondly, how to balance the interests of patients and multiple hospitals is also an interesting research issue. Finally, we can further analyze the impact of telemedicine service process in community hospitals on the service rate of doctors in general hospitals.

Data Availability

All data are included within the article.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants nos. 71671032, 61703220, and 61703290, Fundamental Research Funds for the Central Universities under Grant no. N180408019, China Postdoctoral Science Foundation under Grant no. 2019T120569, and Outstanding Youth Innovation Team Project of Colleges and Universities in Shandong Province under Grant no. 2020RWG011.

References

[1] E. Mantikou, M. A. F. M. Youssef, M. van Wely et al., "Embryo culture media and IVF/ICSI success rates: a systematic review," *Human Reproduction Update*, vol. 19, no. 3, pp. 210–220, 2013.

[2] Y. Fu, H. Wang, J. Wang, and X. Pu, "Multiobjective modeling and optimization for scheduling a stochastic hybrid flow shop with maximizing processing quality and minimizing total tardiness," *IEEE Systems Journal*, vol. 99, pp. 1–12, 2020.

[3] W. J. Zhang and C. A. van Luttervelt, "Toward a resilient manufacturing system," *CIRP Annals*, vol. 60, no. 1, pp. 469–472, 2011.

[4] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, p. 417, 2009.

[5] J. W. Wang, R. L. Dou, R. R. Muddada et al., "Management of a holistic supply chain network for proactive resilience: theory and case study," *Computers & Industrial Engineering*, vol. 125, pp. 668–677, 2017.

[6] Y. Zhang, S. Cheng, Y. Shi, D.-w. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.

[7] Y. Fu, D. Wu, Y. Wang, and H. Wang, "Facility location and capacity planning considering policy preference and uncertain demand under the One Belt One Road initiative," *Transportation Research Part A: Policy and Practice*, vol. 138, pp. 172–186, 2020.

[8] W. J. Zhang, J. W. Wang, and Y. Lin, "Integrated design and operation management for enterprise systems," *Enterprise Information Systems*, vol. 13, no. 4, pp. 424–429, 2019.

[9] H. Ouyang, N. T. Argon, and S. Ziya, "Allocation of intensive care unit beds in periods of high demand," *Operations Research*, vol. 68, no. 2, pp. 591–608, 2020.

[10] B. Denton and D. Gupta, "A sequential bounding approach for optimal appointment scheduling," *IIE Transactions*, vol. 35, no. 11, pp. 1003–1016, 2003.

[11] J. Huang, B. Carmeli, and A. Mandelbaum, "Control of patient flow in emergency departments, or multiclass queues with deadlines and feedback," *Operations Research*, vol. 63, no. 4, pp. 892–908, 2015.

[12] S. A. Erdogan, A. Gose, and B. T. Denton, "Online appointment sequencing and scheduling," *IIE Transactions*, vol. 47, no. 11, pp. 1267–1286, 2015.

[13] S. He, M. Sim, and M. Zhang, "Data-driven patient scheduling in emergency departments: a hybrid robust-stochastic approach," *Management Science*, vol. 65, no. 9, pp. 4123–4140, 2019.

[14] X. Li, J. Wang, and R. Y. K. Fung, "Approximate dynamic programming approaches for appointment scheduling with patient preferences," *Artificial Intelligence in Medicine*, vol. 85, pp. 16–25, 2018.

[15] D. Wang, K. Muthuraman, and D. Morrice, "Coordinated patient Appointment scheduling for a multistation healthcare network," *Operations Research*, vol. 67, no. 3, pp. 599–618, 2019.

[16] R. Atar, A. Goswami, and A. Shwartz, "Risk-sensitive control for the parallel server model," *SIAM Journal on Control and Optimization*, vol. 51, no. 6, pp. 4363–4386, 2013.

[17] N. Liu, S. R. Finkelstein, M. E. Kruk, and D. Rosenthal, "When waiting to see a doctor is less irritating: understanding patient preferences and choice behavior in appointment scheduling," *Management Science*, vol. 64, no. 5, pp. 1975–1996, 2017.

[18] A. K. Dogru and S. H. Melouk, "Adaptive appointment scheduling for patient-centered medical homes," *Omega*, vol. 85, pp. 166–181, 2019.

[19] H.-J. Schütz and R. Kolisch, "Approximate dynamic programming for capacity allocation in the service industry," *European Journal of Operational Research*, vol. 218, no. 1, pp. 239–250, 2012.

[20] V.-A. Truong, "Optimal advance scheduling," *Management Science*, vol. 61, no. 7, pp. 1584–1597, 2015.

[21] R. Hassin and L. Ravner, "Delay-minimizing capacity allocation in an infinite server-queueing system," *Stochastic Systems*, vol. 9, no. 1, pp. 27–46, 2019.

[22] T. R. Rohleder, D. Cooke, P. Rogers, and J. Egginton, "Coordinating health services: an operations management perspective," *Handbook of Healthcare Operations Management*, vol. 184, pp. 421–445, 2013.

[23] S. Saghafian, W. J. Hopp, S. Iravani et al., "Workload management in telemedical physician triage and other knowledge-based service systems," *Management Science*, vol. 64, no. 11, pp. 4967–5460, 2018.

[24] B. Rajan, T. Tezcan, and A. Seidmann, *The Process Implications of Using Telemedicine for Chronically Ill Patients: Analyzing Key Consequences for Patients and Medical Sspecialists*, California State University, Hayward, CA, USA, 2015.

[25] T. H. Westerdale, "An approach to credit assignment in classifier systems," *Complexity*, vol. 4, no. 2, pp. 49–52, 2015.

[26] S. Chen, Q. Qin, J. F. Chen, and Y. H. Shi, "Brain storm optimization algorithm: a review," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 445–458, 2016.

[27] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy,"

IEEE Transactions on Cybernetics, vol. 46, no. 10, pp. 2238–2251, 2016.

[28] H. Lu, R. R. Zhou, S. Cheng, and Y. H. Shi, "Multi-center variable-scale search algorithm for combinatorial optimization problems with the multimodal property," *Applied Soft Computing*, vol. 84, 2019.

[29] Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, "Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 229–242, 2017.

[30] Z. Huang, W. M. P. van der Aalst, X. Lu, and H. Duan, "Reinforcement learning based resource allocation in business process management," *Data & Knowledge Engineering*, vol. 70, no. 1, pp. 127–145, 2011.

[31] D. B. Noureddine, A. Gharbi, and S. B. Ahmed, "Multi-agent deep reinforcement learning for task allocation in dynamic environment," in *Proceedings of the 12th International Conference on Software Technologies*, pp. 17–26, Madrid, Spain, 2017.

[32] Z. Xiao, S. X. Ma, and S. Y. Zhang, "Learning task allocation for multiple flows in multi-agent systems," in *Proceedings of the 2009 International Conference on Communication Software and Networks*, pp. 153–157, Chengdu Sichuan, China, 2009.

[33] A. Asghari, M. K. Sohrabi, and F. Yaghmaee, "Online scheduling of dependent tasks of cloud's workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents," *Soft Computing*, vol. 24, no. 21, pp. 16177–16199, 2020.

[34] T. Wauters, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe, "A learning-based optimization approach to multi-project scheduling," *Journal of Scheduling*, vol. 18, no. 1, pp. 61–74, 2015.

[35] R. S. Sutton and G. A. Barto, *Reinforcement learning: an introduction*, MIT Press, vol. 17, no. 2, p. 322, Cambridge, MA, 1998.

[36] Y. Gao, S. F. Chen, and X. Lu, "Research on reinforcement learning technology: a review," *Acta Automatica Sinica*, vol. 30, no. 1, pp. 86–100, 2004.

Hindawi
Complexity
Volume 2020, Article ID 8883945, 7 pages
https://doi.org/10.1155/2020/8883945

Research Article

A New Method to Construct the KD Tree Based on Presorted Results

Yu Cao, Huizan Wang, Wenjing Zhao, Boheng Duan, and Xiaojiang Zhang

College of Meteorology and Oceanography, National University of Defense Technology, Changsha, Hunan Province, China

Correspondence should be addressed to Huizan Wang; wanghuizan@126.com

Received 25 September 2020; Revised 1 December 2020; Accepted 10 December 2020; Published 23 December 2020

Academic Editor: Shi Cheng

Searching is one of the most fundamental operations in many complex systems. However, the complexity of the search process would increase dramatically in high-dimensional space. K-dimensional (KD) tree, as a classical data structure, has been widely used in high-dimensional vital data search. However, at present, common methods proposed for KD tree construction are either unstable or time-consuming. This paper proposed a new algorithm to construct a balanced KD tree based on presorted results. Compared with previous similar method, the new algorithm could reduce the complexity of the construction process (excluding the presorting process) from O ($KNlog_2N$) level to O ($Nlog_2N$) level, where K is the number of dimensions and N is the number of data. In addition, with the help of presorted results, the performance of the new method is no longer subject to the initial conditions, which expands the application scope of KD tree.

1. Introduction

How to search quickly is one of the most fundamental problems in many research fields [1–3], such as grid remapping, pattern recognition, and ray tracing. However, the complexity of the search problem would increase dramatically in high-dimensional space. Take the basic problem of finding the nearest point for a target point in a high-dimensional space as an example. The most intuitive way to do this is to compute the distance from all the other points, and then pick out the point with the shortest distance. However, the higher the dimension is, the more expensive it is to calculate the distance. Calculating the distances between the target point and all the other points would soon become unacceptable in high-dimensional cases. At this point, an efficient algorithm to avoid unnecessary calculation is particularly important.

KD tree [4] is a classical data structure that stores K-dimensional points for quick retrieval by the form of a binary tree. Different from a standard binary tree, each level of the KD tree can be divided by different dimensions. Due to its good performance in solving multidimensional searching problems, it has been widely used in multidimensional critical data search (e.g., regional search and nearest neighbor search [5, 6]).

There are two vital operations in building a KD tree: one is choosing the dimension to be divided, and the other is selecting the exact splitting point. In terms of choosing the splitting dimension, the dimension with the largest variance or the widest dispersion is generally recommended [7], because such choices can divide the search space more evenly. Nevertheless, for the sake of simplicity, it is acceptable to divide the dimensions in a circular fashion in many cases, especially when the points are evenly distributed. As for selecting the splitting point, due to fact that no effective rebalancing techniques are available for KD tree reconstruction at present [8], the median point is usually suggested to be chosen so as to build a balanced tree directly [9].

A variety of methods, aiming at picking the median point out quickly and accurately, have been developed at present. Quicksort [10] is one of the most popular sorting algorithms, which could find the median in O ($Nlog_2N$) time at best. In addition, an improved method (i.e., quick select algorithm [11]) could even reduce the time to O (N) level, which is regarded as one of the fastest methods. Unfortunately, the

performance of both methods might degrade to O (N^2) level at worst. Though some of other methods (e.g., merge sort [12]) may guarantee to obtain the median in O ($Nlog_2N$) time, their performance is still slightly disappointing.

As we all know, sorting algorithms have been relatively mature at present. Therefore, it should be an intuitive idea to build a KD tree based on presorted results. However, few of previous literature are concerned about this idea. To the best of the authors' knowledge, Brown [13] is the first one who proposed a method to build a KD tree based on presorted results in O ($KNlog_2N$) at worst. Cao [14] further improved the performance of this method by replacing superkeys with keys during the presorting and construction. However, both methods have to maintain K index arrays all the time during the construction, which degrades the complexity of the construction process from O ($Nlog_2N$) to O ($KNlog_2N$). In addition, the selection of the splitting dimension is restricted for the sake of arrays reusing.

This paper proposed a new method to construct the KD tree based on presorted results, which can not only build a KD tree in O ($Nlog_2N$) time under any conditions, but also arbitrarily choose the splitting dimension. Detailed description of the new method is given in Section 2. Experiments' validation and analysis are shown in Section 3. Finally, the conclusion is drawn in Section 4.

2. Algorithm

2.1. Basic Idea. Brown's method would prepare an ordered index array for each partition. Each partition can be implemented quickly according to the index array it depends on. Taking the splitting order adopted by Brown as an example (i.e., split by x, y and z in turn for 3D data), assuming that the index array is arranged from small to large, elements less than the median belong to the left subtree, while elements greater than the median belong the right subtree (this rule will be adopted through this paper). The first partition is based on the x index array. The middle element of the x index array is just the median point. Elements in the upper part of the index array belong to the left subtree, while elements in the lower part of the array belong to the right subtree. The overhead of selecting the median point is only O (1). In order to prepare for the second partition, it needs to form a new y index array after O (N) comparisons. In theory, with the new y index array, the second partition has been able to be carried out successfully. However, z index array is also updated by Brown in the first partition. In fact, this is accomplished for the preparation of the third partition. If z index array is not updated during the first partition, the program will not be able to determine which subtree the element in z index array belongs to in the second partition, which would lead to a failure of forming the new z index array in the third partition. However, such operations lead to the degradation of the complexity of KD tree construction from O ($Nlog_2N$) to O ($KNlog_2N$).

Therefore, in order to build a KD tree in O ($Nlog_2N$) time, only the index array useful for next partition can be maintained during the construction. To achieve this goal, we need to ensure that each element clearly knows which subtree it belongs to in each partition. Hence, we designed three additional integer arrays to record the state of corresponding elements. They are BN array, SS array, and CUR array. The size of each array is N. BN[i] is used to record the starting position of the subtree to which element i belongs. SS[j] is used to record the number of the remaining elements in the subtree of the element whose starting position is j. CUR[k] is used to record the number of elements that have been arranged in the subtree of the element whose starting position is k. If CUR[BN[i]] is less than half of SS[BN[i]], it means that element i should belong to the left subtree of the current subtree. Similarly, if CUR[BN[i]] is larger than half of SS[BN[i]], it means that element i should belong to the right subtree of the current subtree. If CUR[BN[i]] equals half of SS[BN[i]], it means that element i is the median of the current subtree. Then, BN[i] should point to the middle of the domain where the subtree occupies (supposing it points to M), and SS[M] should be set to 0. When CUR[BN[i]] equals SS[BN[i]], it means that all the elements in the current subtree have been processed, and corresponding elements in SS and CUR arrays should be reset so as to prepare for the next split. More specifically, supposing the starting position of the left subtree and the right subtree of the current subtree is L and R respectively, then SS[L] and SS[R] should be set to half of SS[BN[i]], while CUR[L] and CUR[R] should be set to 0.

Let us take building a KD tree for seven 2D points as an example. The coordinate information of these seven points is shown on the left side of Figure 1 under "Tuples." Index arrays presorted by values in the x and y coordinates are listed from small to large and shown under "Presorted Results." The initialization of BN, SS, and CUR arrays is shown under "Initial." BN is initialized to 0, indicating that the data set of the current subtree starts from position 0. SS is initialized to 7, and it means that there are 7 elements in current subtree, and CUR is initialized to 0, indicating that none of the elements in the current subtree have been arranged. It should be noted that, for SS array and CUR array, only SS[0] and CUR[0] are meaningful, and assignments for other elements have no influence on the final result.

After the completion of these preparations, the construction of the KD tree formally begins. For the sake of simplicity, subtrees in each level are divided by x and y sequentially and cyclically. Then, the first partition is accomplished by processing the elements in x index array in turn from top to bottom. Elements 2, 4, and 3 are less than the median, so they belong to the left subtree and should be left in the upper part of the domain occupied by the current subtree (i.e., their corresponding value in the BN array should be 0). Element 0 is the median, so it should be located in the middle of the current subtree (i.e., BN[0] should be 3). Moreover, as this point will no longer belong to any subtree after being an intermediate node of the KD tree, SS[3] should be set to 0. Elements 5, 6, and 1 are greater than the median, so they belong to the right subtree and should be left in the lower part of the domain occupied by the current subtree (i.e., their corresponding value in the BN array should be 4). In the process of traversing the subtree starting from 0, the

	Tuples (x, y)	Presorted results x	y	Initial BN	SS	CUR	After first split BN	SS	CUR	After second split BN	SS	After third split BN	SS	F
0	(6, 1)	2	0	0	7	0	3	3	0	3	1	3	0	2
1	(9, 4)	4	2	0	7	0	4	7	0	5	0	5	0	3
2	(1, 2)	3	6	0	7	0	0	7	0	0	1	0	0	4
3	(3, 5)	0	1	0	7	0	0	0	0	1	0	1	0	0
4	(2, 8)	5	3	0	7	0	0	3	0	2	1	2	0	6
5	(7, 9)	6	4	0	7	0	4	7	0	6	0	6	0	1
6	(8, 3)	1	5	0	7	0	4	7	0	4	1	4	0	5

Figure 1: An example to construct the KD tree based on the new method.

value of CUR[0] increases continuously, which is used to determine whether the current element belongs to the left subtree or the right subtree. After all the elements in the current subtree have been visited, CUR[0] is reset to 0. At the same time, the number of elements in the left and right subtrees will also be reset; that is, SS[0] and SS[4] will be set to 3. After the first partition, the values of these three arrays are shown in Figure 1 under "After First Split." It can be seen that the CUR array is an auxiliary array, which only affects the current partition. After the current partition is completed, it will return to 0 again. Therefore, its state will no longer be displayed in the subsequent partition process. The second partition will be accomplished according to the y index array. The first element in y index array is 0, and the beginning position of the current element (i.e., BN[0]) is 3. SS[3] is zero, which means that the current element does not need to be processed. The second element is 2, the starting position of element 2 (i.e., BN[2]) is 0, and the number of elements in the current subtree is 3. Therefore, element 2 belongs to the left subtree of the current subtree, and BN[2] should be 0. The third element is 6, starting at 4 and belonging to the left subtree of the current subtree, so BN[6] is 4. The fourth element is 1, and the starting position is 4, which should be the median of the current subtree. Therefore, BN[4] is set to 5, and SS[5] is set to 0. The remaining elements can be processed similarly. The results are shown in Figure 1 under "After Second Split." When all the elements are picked out as a node in the KD tree, all the elements in the SS array will be zero, as shown under "After Third Split" in Figure 1. Then, the final index array (i.e., F array in the right side of Figure 1) used to construct the KD tree can be easily obtained from the BN array (it can be done by placing i in the position of BN[i] of F array). It can be seen from the process descripted above that when building a KD tree, the new method does not manage to prepare a complete index array for each partition, but only maintains the set of data to be partitioned next time. This idea enables the new method no longer need to update K index arrays every time. Thus, it reduces the time cost.

It should be noted that we have just obtained the final index array for construction, not the final KD tree. Previous algorithms use recursive method to build KD tree, so the child node can easily link back to the parent node. However, the previous process no longer uses recursion, so it is impossible to transfer the information of the child node to the parent node, unless more space is used to store the information of the parent node. Fortunately, after obtaining the final index array F, we can visit the index array again according to the same rules and order recursively as before to build the KD tree. The difference is that the current recursive traversal process no longer needs to find the median point but only needs to get the middle point from the array range occupied by the subtree. It should be noted that if the selection of the split dimension does not follow some fixed rules, additional arrays are needed to record the selection results of the split dimension before building the KD tree recursively at last.

2.2. Detailed Description. The detailed description of building a KD tree based on the new method is shown in Figure 2. The first step is presorting and initialization, after that, choosing the splitting dimension and processing BN, SS, and CUR arrays repeatedly, until each element in the SS array is zero. Subsequently, the final index array F is constructed according to the BN array (in fact, the construction of F array can also be accomplished during the partition without the help of BN array). Finally, the KD tree is exactly constructed recursively according to the F array. The main process of dealing with BN, SS, and CUR arrays is shown in the large box. Elements in the D index array (i.e., the splitting dimension) will be visited in turn from the beginning to the end. Suppose that the current element is tmpi. If the current element has already been selected (i.e., tmpsize is 0), it no longer needs to be processed. Otherwise, it is whether it belongs to the left subtree or the right subtree or is the median will be judged according to CUR[tmpBN]. If it belongs to the left subtree or the right subtree, BN[tmpi] will point to the starting position of the corresponding left or right subtree. Otherwise, BN[tmpi] will point to the middle of the segment occupied by the current subtree, and its corresponding value in the SS array will be set to zero. When all the elements in the current subtree have been processed, the corresponding elements in SS and CUR arrays will be reset as described in the foregoing subsection.

The process of building a KD tree recursively according to the final index array F is shown in Figure 3. The first step is to obtain the splitting dimension. The splitting dimension

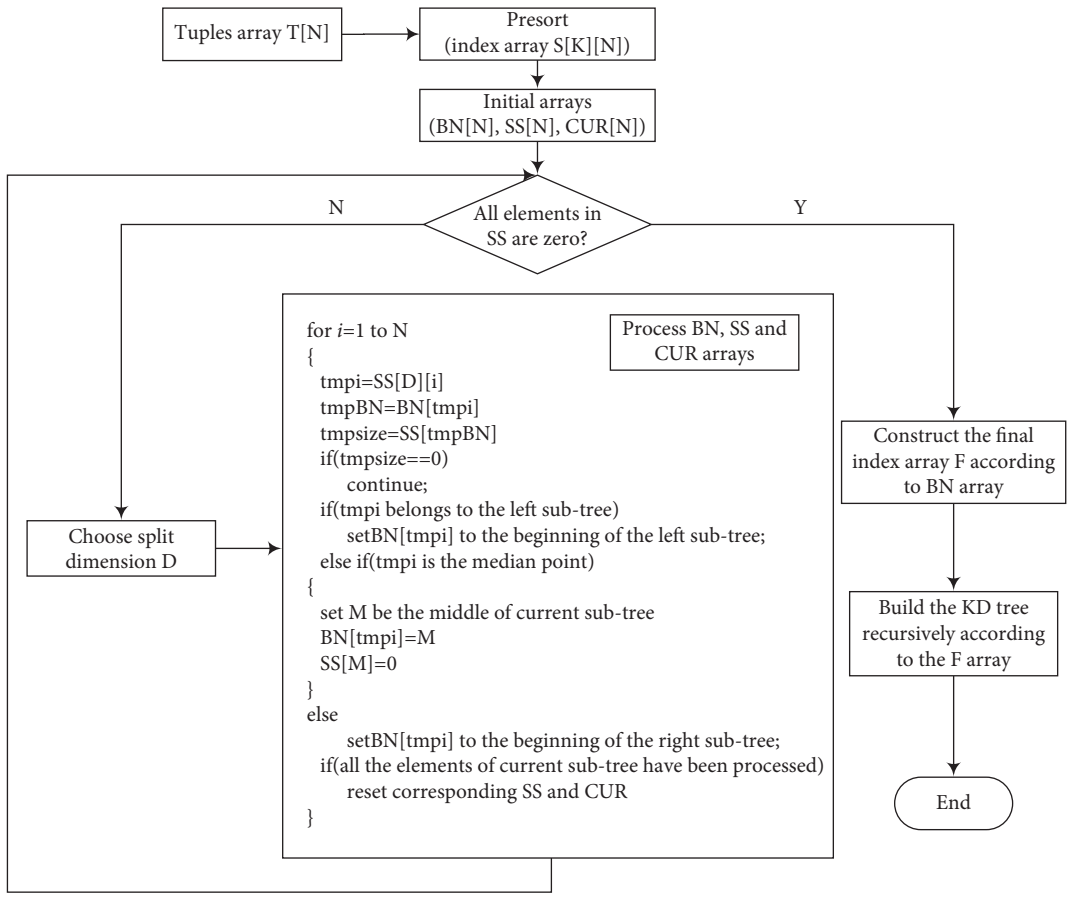


Figure 2: Building a KD tree based on the new method.

can be obtained either by following the same rules or from the record of previous partition results. The next step is to select the median point. In fact, the middle element of the segment occupied by current subtree is just what we want. Subsequently, the starting position and the number of elements for the left and right subtrees will be calculated, respectively. Finally, the left and right subtrees will be constructed recursively until the number of elements in the current subtree is zero.

2.3. Complexity Analysis. From the foregoing discussion, it is known that the whole process of building a KD tree is comprised of two parts. The first part is to form the final index array for construction, while the second part is to exactly build the KD tree according to the final index array. In the first process, one of the presorted index arrays will be traversed during each partition. Although each element might be accessed in a different order, the final result is that all the elements will be processed once. Therefore, the time complexity of each partition is O (N). There are O ($\log_2$N) times of partition in need, so the time complexity for the first process is O (N$\log_2$N) in total. It is worth noting that although each partition will access some selected nodes, the overhead is limited, and the impact on the final execution time is negligible. In the second process, each selection of the KD node is accomplished in O (1) time. Therefore, it takes O (N) time to build the KD tree. Adding up the complexity of these two parts, the final time complexity of the new method is O (N $\log_2$N + N). When N is large, the total time complexity can be considered as O (N$\log_2$N). It is notable that the time complexity of the new method is independent of the initial order of original elements. It means that, even under the worst condition, the time complexity of the new method is still O (N$\log_2$N) level, which is just what we expected.

3. Experiments

The new method is accomplished in C language. As the presorting process is not the focus of this paper, for the sake of simplicity, the function "qsort", which is provided by the C standard library, is used for presorting. There are two data sets used for testing. One owns 2^{24} 6-dimensional real elements, which are randomly generated between 0 and 100 with 6 valid decimal places. The other owns 2^{17} 6-dimensional real elements with the same range. The main difference between these two data sets is that the elements in the latter data set are arranged from large to small in each dimension.

Figure 4 shows the construction time (seconds) for $2^{18} \le N \le 2^{24}$ 4-dimensional randomly generated real elements based on the new method and the improved method (more details about the improved method can be seen in [14]). Since the execution time of these two methods is

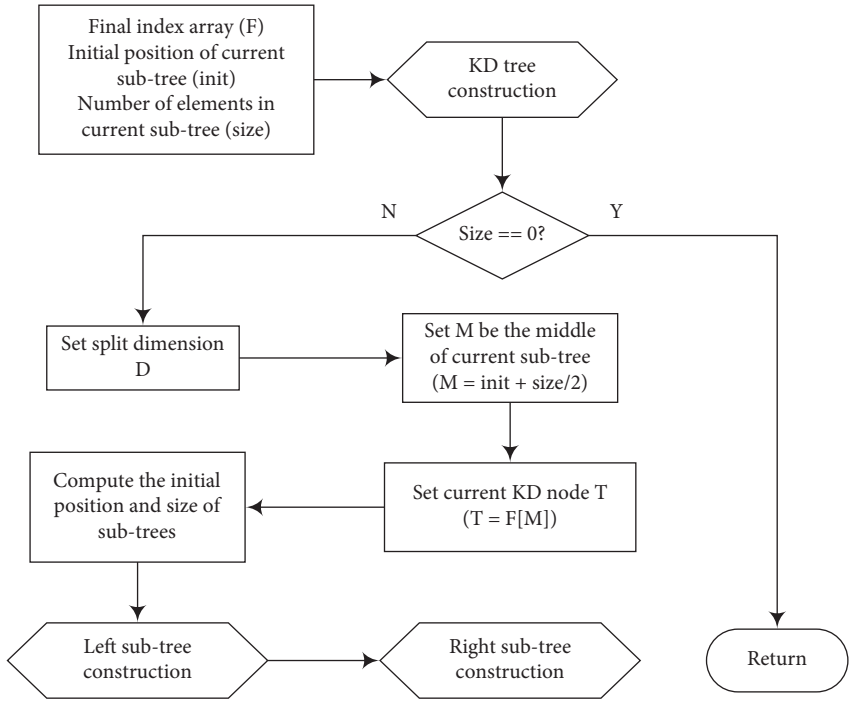


Figure 3: Building a KD tree recursively according to the final index array.

supposed to be proportional to N $\log_2$N, the x-axis expands by a factor of $\log_2$N, so that the result can be fitted to two straight lines. The construction time based on the new method is marked by circles and fitted as the solid line, while construction time based on the improved method is marked by diamonds and fitted as the dashed line. It can be seen clearly from the picture that the execution time of these two methods increases almost linearly with N$\log_2$N indeed, which demonstrates the correctness of our analysis about the complexity of the new method. What is more, the performance of the new method is always better than the improved method in the current case, which is exactly what we expected. In addition, the overhead of the construction grows a little bit faster when the number of elements is too large, and this might be due to the increase of Cache failure caused by large scale data.

In order to further verify the performance of the new method in different dimensions, we carefully compare the new method with the improved method and the quick select method (more details about the process of the quick select method can be found in [15]). Since the construction time of all the three methods tends to grow linearly with the increase of the dimension, the results for these three methods are also fitted to straight lines (see Figure 5). It is pleasing to find that the new method has almost equivalent performance with the quick select method. Though the execution time based on the new method is still slightly larger than that based on the quick select method, considering that the quick select method is the best at handling random data, the new approach has performed well enough. In addition, the improved method performs equivalently in our 3D case and better in our 2D case, and this is because the improved method does not maintain unnecessary index arrays when

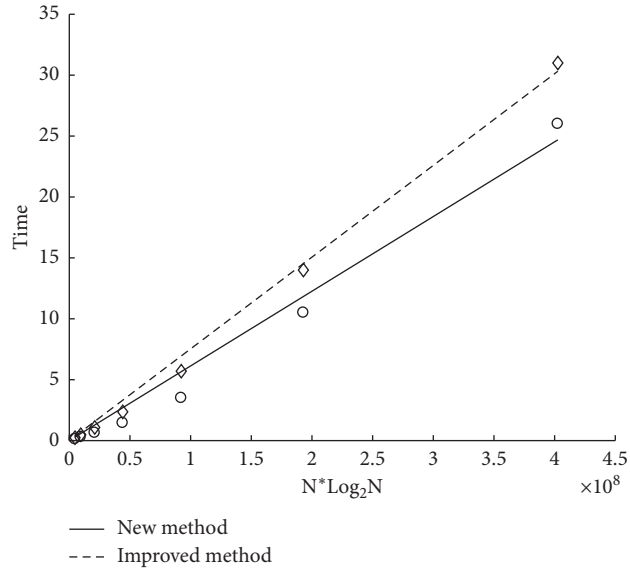


Figure 4: Construction time (seconds) for $2^{18} \le N \le 2^{24}$ 4-dimensional randomly generated real elements.

processing 2D data. Meanwhile, the operation adopted by the improved method is simpler. The disadvantage of maintaining all index arrays in the improved method is only shown when dealing with high-dimensional problems.

The quick select method adopted in this paper always chooses the first element as the pivot element to partition the remaining subarray. Therefore, its performance in each recursion would degrade to O (N^2) level when the arrays are arranged from large to small. Figure 6 shows the construction time (seconds) for 2^{17} real elements (the elements

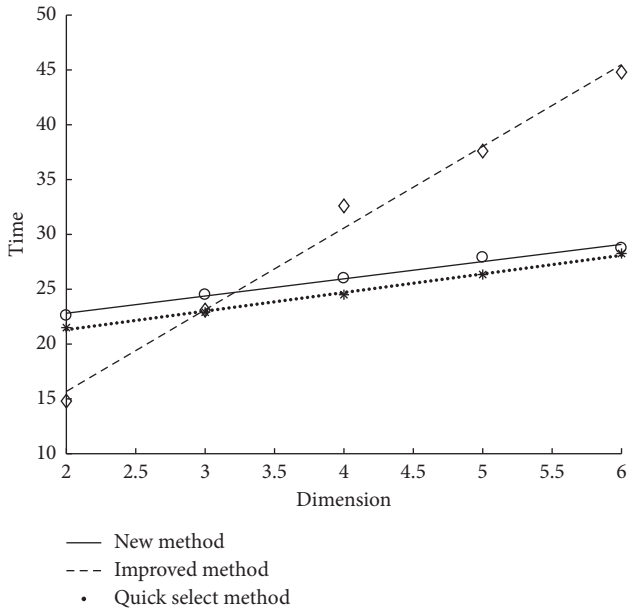


Figure 5: Construction time (seconds) for 2^{24} randomly generated real elements in different dimensions.

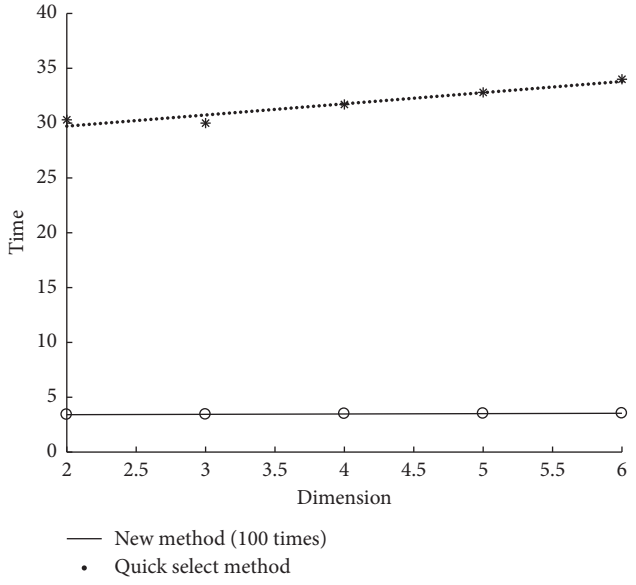


Figure 6: Construction time (seconds) for 2^{17} real elements (the elements are arranged from large to small in each dimension) in different dimensions. The execution time based on the new method is magnified by 100 times.

are arranged from large to small in each dimension) in different dimensions. As the construction time based on the new method is approaching zero, the result of the new method is magnified by 100 times in this figure. Comparing with Figure 5, it can be seen that the execution time based on the quick select method in Figure 6 even exceeds the time cost of processing 2^{24} elements in Figure 5. In contrast, the performance of the new method is quite stable, which performs far better than the quick select method. It illustrates that the performance of the new method is not affected by the initial conditions of the data.

4. Conclusions

In this paper, we proposed a new method that guarantees to construct the KD tree in O ($Nlog_2N$) time (excluding the overhead of presorting process), with the help of three additional integer arrays. Compared with previous methods, the new method has almost equivalent performance with that based on the quick select method for random data and performs much better under extreme conditions. Though the improved method [14] performs better in two dimensional cases, the benefits of the new method soon become apparent in high-dimensional cases.

The new method is suitable for complex systems that need multidimensional queries of massive data. For example, in a cloud storage system based on key-value pair model, multidimensional queries often require a complete scan of the entire data set, which is very inefficient. With the help of KD tree, the efficiency of the query would be improved greatly [16], and our method will improve the efficiency of building the KD tree significantly.

In order to further reduce the execution time of KD tree construction, parallelism is essential [17–20]. Construction method based on recursion usually carries out the construction of the left and right subtrees in parallel. However, the main part of the new method no longer employs the recursive scheme. Therefore, discovering the characteristics of the data structure and the process of the new method so as to develop new parallel algorithms is the focus of our future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Our previous work about the improved method has been published in "2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)" [14]. In this paper, the authors proposed a new method to construct the KD tree without continuously updating the presorted index arrays, which performs much better than the improved method in high-dimensional cases. The authors would like to thank the funding from the National Key R&D Program of China (2018YFC1406202) and the National Natural Science Foundation of China (41830964).

References

[1] W.-F. Hou, D.-W. Li, C. Xu et al., "An advanced k nearest neighbor classification algorithm based on kd-tree," in *Proceedings of the 2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pp. 902–905, Chongqing, China, December 2018.

[2] B. Choi, R. Komuravelli, V. Lu et al., "Parallel SAH k-D tree construction," in *Proceedings of the Conference on High Performance Graphics (HPG '10)*, pp. 77–86, Saarbrucken, Germany, June 2010.

[3] M. Shevtsov, A. Soupikov, and A. Kapustin, "Highly parallel fast KD-tree construction for interactive ray tracing of dynamic scenes," *Computer Graphics Forum*, vol. 26, no. 3, pp. 395–404, 2007.

[4] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[5] B. Xiao and G. Biros, "Parallel algorithms for nearest neighbor search problems in high dimensions," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. 667–699, 2016.

[6] Md. M. A. Patwary, N. R. Satish, N. Sundaram et al., "PANDA: Extreme scale parallel K-nearest neighbor on distributed architectures," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 494–503, Chicago, IL, USA, May 2016.

[7] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.

[8] G. Adelson-velskii and E. Landis, "An algorithm for the organization of information," *Proceedings of the USSR Academy of Sciences*, vol. 146, pp. 263–266, 1962.

[9] P. K. Agarwal, K. Fox, K. Munagala et al., "Parallel algorithms for constructing range and nearest-neighbor searching data structures," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium*, San Francisco, CA, USA, June 2016.

[10] C. A. R. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest et al., *Introduction to Algorithms*, The MIT Press, Cambridge, MA, USA, 3rd edition, 2009.

[12] J. Zhang, Y.-P. Gao, Y.-S. He et al., "Algorithm improvement of two-way merge sort based on OpenMP," *Applied Mechanics and Materials*, vol. 3682, no. 1403, pp. 24–29, 2015.

[13] R. A. Brown, "Building a Balanced k-d Tree in O(kn log n) Time," *Journal of Computer Graphics Techniques*, vol. 4, pp. 50–68, 2015.

[14] Y. Cao, X.-J. Zhang, B.-H. Duan et al., "An improved method to build the KD tree based on presorted results," ", in *Proceedings of the 11th International Conference on Software Engineering and Service Science*, pp. 71–75, Beijing, China, October 2020.

[15] Y. Cao, B. Wang, W.-J. Zhao et al., "Research on searching algorithms for unstructured grid remapping based on kd tree," in *Proceedings of the 3rd International Conference on Computer and Communication Engineering Technology*, pp. 29–33, Beijing, China, August 2020.

[16] J. He, Y. Wu, F. Yang et al., "Multi-dimensional cloud index based on KD-tree and R-tree," *Journal of Computer Applications*, vol. 34, no. 11, pp. 3218–3221, 2014.

[17] G. D. Fatta and D. Pettinger, "Dynamic load balancing in parallel kd-tree K-means," in *Proceedings of the 2010 IEEE 10th International Conference on Computer and Information Technology*, pp. 2478–2485, Bradford, UK, June 2010.

[18] L. Hu, S. Nooshabadi, and M. Ahmadi, "Massively parallel KD-tree construction and nearest neighbor search algorithms," in *Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS 2015)*, vol. 5, pp. 2752–2755, Lisbon, Portugal, May 2015.

[19] D. Wehr and R. Radkowski, "Parallel kd-tree construction on the GPU with an adaptive split and sort strategy," *International Journal of Parallel Programming*, vol. 46, no. 6, pp. 1139–1156, 2018.

[20] M. Cheng, "Parallel SAH based KD tree construction algorithm," *Advanced Materials Research*, vol. 433-440, pp. 3543–3547, 2012.

Hindawi
Complexity
Volume 2020, Article ID 7264396, 12 pages
https://doi.org/10.1155/2020/7264396

Research Article

BurstBiRank: Co-Ranking Developers and Projects in GitHub with Complex Network Structures and Bursty Interactions

Dengcheng Yan,[1] **Zhen Shao**,[2] **Yiwen Zhang**,[2] **and Bin Qi**[2]

[1]*Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China*
[2]*School of Computer Science and Technology, Anhui University, Hefei 230601, China*

Correspondence should be addressed to Yiwen Zhang; zhangyiwen@ahu.edu.cn

Received 31 July 2020; Accepted 4 December 2020; Published 16 December 2020

Academic Editor: Shi Cheng

With the wide adoption of social collaborative coding, more and more developers participate and collaborate on platforms such as GitHub through rich social and technical relationships, forming a large-scale complex technical system. Like the functionalities of critical nodes in other complex systems, influential developers and projects usually play an important role in driving this technical system to more optimized states with higher efficiency for software development, which makes it a meaningful research direction on identifying influential developers and projects in social collaborative coding platforms. However, traditional ranking methods seldom take into account the continuous interactions and the driving forces of human dynamics. In this paper, we combine the bursty interactions and the bipartite network structure between developers and projects and propose the BurstBiRank model. Firstly, the burstiness between each pair of developers and projects is calculated. Secondly, a weighted developer-project bipartite network is constructed using the burstiness as weight. Finally, an iterative score diffusion process is applied to this bipartite network and a final ranking score is obtained at the stationary state. The real-world case study on GitHub demonstrates the effectiveness of our proposed BurstBiRank and the outperformance of traditional ranking methods.

1. Introduction

Social collaborative coding is now a popular paradigm among software developers, and collaborations of developers from all over the world can be easily conducted with the social and technical functionalities provided by such kind of platforms like GitHub. For example, in GitHub, developers can follow each other to form a social network, keep track of the updates of a project by the star and watch functionalities, contribute codes by the commit and pull request functionalities, or participate in the discussions of new features design or bug fix by the issue functionality. Rich social and technical functionalities connect developers and projects to form a large-scale complex technical system. It is known that critical nodes usually play important role in operation management and optimization of complex systems. The same goes for complex technical systems such as GitHub, which is usually driven by influential developers and projects to more optimized states with higher efficiency for software development. For example, in addition to direct collaboration, developers always seek popular developers and projects for improving coding ability and technical selection, which in turn makes collaborations more efficient. Thus, identifying influential developers and projects is of great significance for the improvement of developer's ability and the prosperity of open source community and also has important applications in service recommendations [1, 2] and quality of service prediction [3–6].

Existing work on influence analysis in open source software community often simply employs basic properties [7], network structural metrics [8], or traditional unipartite graph ranking model [9–12]. The influence of developers

and projects, two major and tightly coupled components of open source software community, is usually evaluated separately although many new graph ranking methods for complex network structures like bipartite network [13] have been proposed. On the other hand, abundant activities of developers are not utilized effectively while our previous study [14] indicates the statistical characteristics of developers' behavior is useful for distinguishing elite and common developers. Figure 1 shows a comparison of contributions between an elite developer Taylor Otwell and a common developer Franz Liedke, which shows different statistical characteristics of their behavior.

In this paper, we aim at mutually identifying influential developers and projects in GitHub by adopting a combination of the burstiness behavior of developers and the bipartite network topology of developer-project interactions. The contributions of this paper are listed as follows:

(1) We propose a burstiness-weighted bipartite network model to incorporate bursty behaviors between developers and projects into network topology.

(2) We combine the diffusion-based ranking method BiRank and the burstiness-weighted bipartite network and propose a new ranking method called BurstBiRank for mutually identifying influential developers and projects in GitHub.

(3) We apply the proposed model to a real-world GitHub dataset, showing that burstiness can correctly measure developers' attention to projects and our model outperforms baseline models.

The remainder of the paper is organized as follows. Section 2 introduces the related works on graph ranking methods and human dynamics. The details of our proposed BurstBiRank method are illustrated in Section 3. Then, the experiment results and discussions are given in Section 4. Finally, we briefly summarize our work and explain future directions in Section 5.

2. Related Work

Influential node identification has been a hot topic in network science research for decades, and many graph ranking methods have been proposed from different views of network structures or information diffusion mechanisms on various kinds of complex networks [15–18]. PageRank [19] and HITS [20] are the most popular ones. PageRank [19] is a random walk-based ranking method and uses the probability a random surfer appears on a web page as the influence score of the web page, while HITS [20] distinguishes authority and hub features of a web page and ranking a web page with both authority score and hub score.

Many graph ranking methods are based on PageRank and HITS. Considering individual's preference, Haveliwala et al. [21] proposed a personalized PageRank algorithm, and a personalized vector was introduced for expressing individual's preference for certain topics, novelty, and sensitivity of individuals' generated contents. Inspired by the discrete-time Markov process interpretation of PageRank, Liu et al. [22] proposed BrowseRank based on continuous-time Markov processes and used user behavior data to rank the importance of pages. In order to overcome parameter tuning of PageRank which is caused by dangling nodes in the network, Lu et al. [23] introduced a ground node connecting to all other nodes and proposed LeaderRank. Then, Li et al. [24] extended LeaderRank to weighted network.

In addition to ranking on unipartite network, recent research studies also extend graph ranking methods of unipartite network to bipartite network. In contrast to random walk-based graph ranking methods, He et al. [13] proposed BiRank, an optimization based ranking method for bipartite network. Xu et al. [25] applied singular value decomposition to bipartite network and proposed SVDRank and SVDARank. Morone et al. [26] extended the k-core decomposition method to the bipartite network and pointed out that in the ecological symbiosis network, the extinction of the maximum k-core node would make the ecosystem reach the critical point of collapse.

The rapid development of graph ranking models also promotes the research in influence analysis for open source software community. Xuan et al. [9] modeled the communications between developers in Apache as networks and analyzed developers' influence using degree, PageRank, and HITS. Joblin [8] et al. classified developers into core and peripheral with several network metrics. From the view of software projects, Inoue et al. [11] constructed a component graph with use relations for ranking software components. Pan et al. [27] constructed a multilayer complex network by extracting structural information from Java software systems and proposed a weighted PageRank algorithm for ranking classes or packages.

Although network structure plays an important role in identifying influential nodes in online social network, based on previous human dynamics study, Yan et al. [14] found that bursty behavior is a good indicator for distinguishing influential developers from common ones. Human dynamics studies the statistical characteristics of spatial or temporal behaviors of human beings and the potential laws behind it. Goh et al. [28] proposed the burstiness metric to measure to which extent the behavior deviates from periodic behavior.

3. Method

In this section, we will present a novel bipartite network ranking framework incorporating burstiness interactions, called BurstBiRank, for mutually identifying influential developers and projects in GitHub. First, we will introduce the definition of burstiness, which plays an important role in our proposed method for measuring how much attention a developer pays on a project. Then, a thorough description is given about the definition and construction of the burstiness-weighted developer-project bipartite network, and a diffusion-based ranking process is applied on this bipartite network. Finally, the overall algorithm is proposed and its time complexity is analyzed. The notations we will use throughout the article are summarized in Table 1.

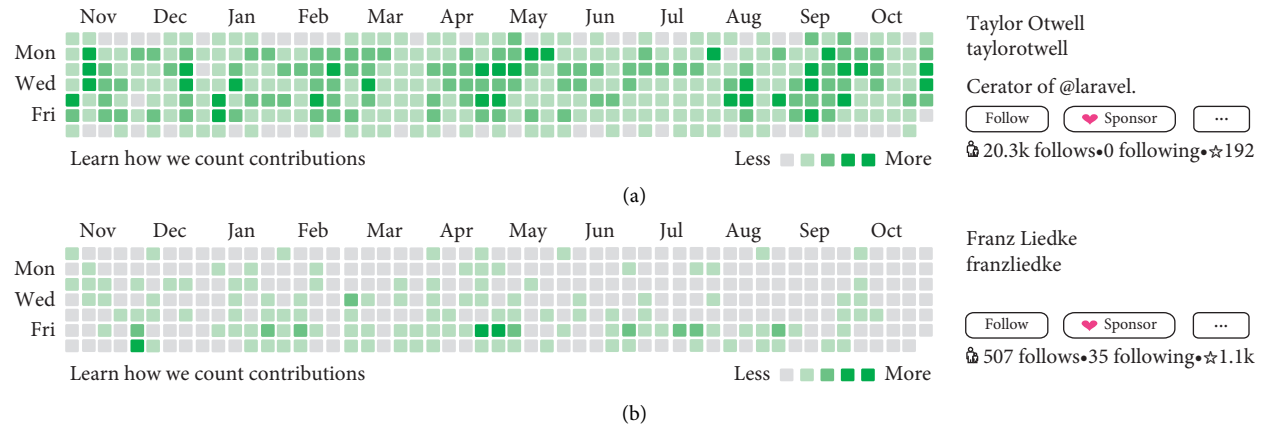


Figure 1: A comparison of two developer behaviors.

Table 1: Notations and explanations.

Symbol	Explanation
u_i	Ranking score of developer i
p_j	Ranking score of project j
u_i^0	Query vectors for developer i
p_j^0	Query vectors for project j
d_i^u, D^u	d_i^u degree of developer i D^u diagonal matrix with $D_{ii}^u = d_i^u$
d_j^p, D^p	d_j^p degree of project j D^p diagonal matrix with $D_{jj}^p = d_j^p$
B_{ij}	Burstiness of interactions between developer i and project j
w_{ij}, W	Weight matrix of burstiness-weighted bipartite network; w_{ij} is its element
α, β	Infection rate and recovery rate in SIR simulation, respectively
γ, λ	Hyperparameters for balancing diffusion scores and prior beliefs
S	Symmetric normalization of weight matrix W

3.1. Burstiness. In many real-world or online systems, people's activity is often intermittent, displaying intense activity during a short period followed by a long period of reduced activity or even no activity. For example, you may spend a total afternoon searching YouTube for videos about husky when you are free, but then seldom visit YouTube in weekdays. This pattern of human behavior and the laws behind it have been studied extensively in the field of human dynamics, and Goh et al. [28] proposed the burstiness metric to measure to which extent the behavior deviates from periodic behavior, which is defined as

$$B = \frac{\sigma_\tau - m_\tau}{\sigma_\tau + m_\tau}, \tag{1}$$

where $\sigma_\tau$and m_τ represent the standard deviation and the mean of the time interval series of human activities, respectively. It can be concluded from the definition: (1) the value of B ranges from -1 to 1; (2) $B > 0$ indicates the behavior is bursty, and the larger B is, the stronger the burstiness is; (3) $B < 0$ indicates cyclical trend, and the smaller B is, the stronger the periodicity is.

3.2. Burstiness-Weighted Developer-Project Bipartite Network

Definition 1. Burstiness-weighted developer-project bipartite network: a burstiness-weighted developer-project bipartite network is a weighted bipartite network $G = (U \cup P, E)$, where U and P denote two disjoint sets of nodes, that is, set of developers and set of projects, respectively, and E represents edges between developers and projects. The burstiness-weighted developer-project bipartite network can be described by a bipartite weight matrix W ($\in R^{|U|\times|P|}$) with elements w_{ij} ($1i \le |\mathrm{U}|$, $1j \le |\mathrm{P}|$) indicating tie strength between developer i and project j, which is a function of the burstiness of interactions between developer i and project j, that is,

$$w_{ij} = f\left(B_{ij}\right). \tag{2}$$

Figure 2 shows a sample burstiness-weighted developer-project bipartite network. Multiple interactions between each pair of developer and project are grouped and burstiness is calculated first. Then, the bipartite weight matrix W is constructed using equation (2). Essentially, the function in equation (2) can be any form, linear or nonlinear. According to the characteristics of burstiness, to ensure edge weights are positive and cyclical interactions have larger weight, we choose a linear form of function f shown in equation (3) for simplicity.

$$w_{ij} = f\left(B_{ij}\right) = -B_{ij} + 1. \tag{3}$$

3.3. BiRank. Score diffusion is a general idea behind many popular ranking methods as PageRank [19] and BiRank [13], which employ an iterative process of diffusing score to neighbors until the stationary state. The final scores at stationary state are regarded as the ranking scores. The process of score diffusion can be formulized as equations (4) and (5), and scores of developers and projects are updated in turns.

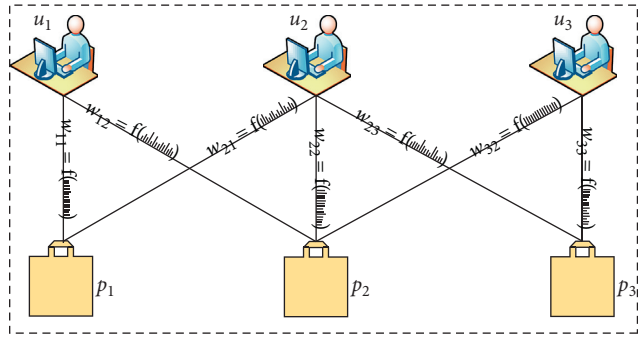


Figure 2: Burstiness-weighted developer-project bipartite network. The edge weight is function of burstiness which can be calculated using interaction sequences.

$$u_i = \sum_{j=1}^{|P|} w_{ij} p_j, \tag{4}$$

$$p_j = \sum_{i=1}^{|U|} w_{ij} u_i. \tag{5}$$

To ensure the convergence and stability, BiRank adopts symmetric normalization.

$$\begin{aligned} u_i &= \sum_{j=1}^{|P|} \frac{w_{ij}}{\sqrt{d_i^u}\sqrt{d_j^p}} p_j, \\ p_j &= \sum_{i=1}^{|U|} \frac{w_{ij}}{\sqrt{d_i^u}\sqrt{d_j^p}} u_i. \end{aligned} \tag{6}$$

BiRank also adopts a query vector in the ranking model to utilize prior beliefs on rankings of nodes as shown in equations (7) and (8). Prior beliefs and diffusion scores are balanced with hyperparameters γ and λ for developers and projects, respectively.

$$u_i = \gamma \sum_{j=1}^{|P|} \frac{w_{ij}}{\sqrt{d_i^u}\sqrt{d_j^p}} p_j + (1-\gamma) u_i^0, \tag{7}$$

$$p_j = \lambda \sum_{i=1}^{|U|} \frac{w_{ij}}{\sqrt{d_i^u}\sqrt{d_j^p}} u_i + (1-\lambda) p_j^0. \tag{8}$$

Finally, the equivalent matrix form of BiRank [7] can be obtained as equations (9) and (10)

$$u = \gamma S p + (1-\gamma) u^0, \tag{9}$$

$$p = \lambda S^T u + (1-\lambda) p^0, \tag{10}$$

where S is the symmetric normalization of weight matrix W.

$$S = \left(D^u\right)^{(-1/2)} W \left(D^p\right)^{(-1/2)}. \tag{11}$$

3.4. Overall Algorithm. Combing Sections 3.1, 3.2, and 3.3, we finally propose the BurstBiRank, and the overall algorithm is shown in Algorithm 1.

3.5. Time Complexity Analysis. The time complexity of BurstBiRank consists of two parts. The first part is the calculation of time intervals, burstiness, and edge weights, so the time complexity is $O(N_I + 2 * N_G)$, where N_I is the number of interactions and N_G is the number of developer-project groups. The second part is the iterative process of BurstBiRank algorithm, and the time complexity of equations (9) and (10) is $O(|U| \cdot |P|)$. However, most real-world networks are usually very sparse and only nonzero elements (which correspond to existing edges) should be stored and computed regarding matrix multiplication of $S^T u$ and Sp. Thus, the time complexity of the second part is $O(c|E|)$, where c is the number of iterations and $|E|$ is the number of edges. The overall time complexity of BurstBiRank is $O(N_I + 2 * N_G + c|E|)$.

4. Experiment

In this section, the performance of BurstBiRank is evaluated against real-world GitHub dataset [29]. All experiments are run on a Windows 10 PC with a corei7-4790 3.6 GHz CPU and 16 GB memory.

4.1. Datasets. GHTorrent dataset is an offline mirror of data offered through the GitHub REST API and a subset of it about PHP development community is used in this experiment. GHTorrent dataset as of November 1, 2018, is selected and preprocessed as follows: (1) commit interactions between developers and PHP projects are selected; (2) commit date is extracted from commit timestamp; (3) multiple commit interaction records of the same date are merged as one record; (4) developers who have equal or less than 10 records are excluded; (5) follow relationship between developers and watch interactions between developers and projects are extracted. The statistics of the dataset after preprocessing are shown in Table 2.

Input:
Developer-project interaction set (DP); query vectors u^0, p^0; and hyperparameters γ, λ
Output:
Ranking vectors u, p;
(1) Group developer-project interactions by developer and project;
(2) **for** developer-project interactions group in all groups **do**
(3) Sort developer-project interactions by commit time in descending order;
(4) Calculate time intervals between successive records;
(5) Calculate burstiness B_{ij};
(6) Calculate edge weight w_{ij} according to equation (3);
(7) **end for**
(8) Construct weight matrix W;
(9) Symmetrically normalize W according to equation (11);
(10) Randomly initialize u and p;
(11) **while** Stopping criteria are not met **do**
(12) Update u and p in turn according equations (9) and (10);
(13) **end while**
(14) **return** u and p

Algorithm 1: BurstBiRank algorithm.

Table 2: The statistics of the dataset.

Data	Count
Developers	129649
Projects	171296
Follows	41881
Watchers	27154
Developer-project commit interactions	8327263

4.2. Evaluation Metrics. Correlation analysis and SIR (susceptible-infected-removed) simulation are usually adopted for evaluation of graph ranking methods.

In correlation analysis, ranking results are compared with the ground truth using correlation coefficients. Kendall's tau [30] is one of such correlation coefficients and compares ranking orders instead of exact ranking scores or ground truth values. The definition of Kendall's tau is shown in the following equation:

$$\tau(X, Y) = \frac{2(C - D)}{n(n-1)}. \tag{12}$$

X and Y are two different lists with length n, which are usually the predicted ranking list and the ground truth ranking list. C and D are the numbers of concordant and discordant pairs between X and Y, respectively. Let $x_i, x_j \in X$and $y_i, y_j \in Y$; if $\text{sign}(x_i - x_j)\text{sign}(y_i - y_j) > 0$, then (x_i, y_i) and (x_j, y_j) are called a concordant pair, and if $\text{sign}(x_i - x_j)\text{sign}(y_i - y_j) < 0$, then (x_i, y_i) and (x_j, y_j) are called a discordant pair. In case of $\text{sign}(x_i - x_j)\text{sign}(y_i - y_j) = 0$, the pair is neither concordant nor discordant.

For the complexity of measuring influence, the ground truth for correlation analysis uses simply the degree of developer-developer following network or developer-project watching network. As we know, degree is local centrality metric which can only roughly measure node's influence from a local view while influence of a node in network mainly relates to its ability of spreading information to the whole network. Generally, a node with higher influence will spread information to more nodes in a network. Thus, we adopt the SIR model [31], a classical epidemic model, in our experiment to evaluate the performance of our proposed ranking method. As shown in Figure 3, in the SIR model, nodes in a network have three statuses, that is, susceptible (S), infected (I), and removed (R). Initially, a node is selected as infected node and the others are susceptible nodes. Then, an iterative transmission process is applied. At each iteration, infected nodes infect one of its susceptible neighbors with the probability α, and infected nodes recover to removed status with the probability β. The iterative transmission process stops when there are no infected nodes in the network. The final number of recovered nodes can be regarded as the influence of the node which is initially selected as infected.

4.3. Baseline Methods. To show the effectiveness of our proposed BurstBiRank, we compare it with several baseline ranking methods. In addition to burstiness-weighted developer-project bipartite network, two other developer-project bipartite networks are also constructed with unweighted edge (UW) and commit number-weighted edge (CN), and all baseline methods are evaluated on these two bipartite networks with corresponding suffix such as PageRank-UW. The hyperparameters of BurstBiRank γ and λ are both set to 0.85, and the query vectors u_0 and p_0 are set to the degrees of developers and projects over total number of nodes of each type, respectively, which can be calculated using equations (13) and (14).

$$u_i^0 = \frac{d_i^u}{|U|}, \tag{13}$$

$$p_j^0 = \frac{d_j^p}{|P|}. \tag{14}$$

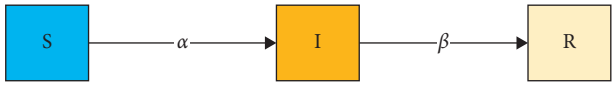


Figure 3: SIR model (source: figure is adapted from Pastor-Satorras et al. [32]).

BiRank [13] employs a diffusion-based way of utilizing mutual reinforcement between different types of nodes in bipartite network for mutually ranking two different types of nodes and adopts a normalization strategy in the iterative process. The hyperparameters are both set to 0.85 in our experiment.

SVDRank and SVDARank [25] apply singular value decomposition to bipartite network and select the first eigenvector as the final ranking vector. The difference between them is that SVDRank runs on the original bipartite network while SVDARank introduces two ground nodes to the original bipartite network before applying singular value decomposition in order to solve the problem of dangling nodes.

PageRank [19] regards ranking in network as a score diffusion process and ranks nodes by iteratively diffusing scores on the network. In our experiment, we directly apply PageRank on the developer-project bipartite networks ignoring types of nodes. The hyperparameter is set to 0.85 in our experiment.

4.4. Results

4.4.1. Correlation Analysis. In the experiment of correlation analysis, the number of followers of a developer in developer-developer following network is chosen as the ground truth for the rankings of developers, and the number of watchers of a project in developer-project watching network is chosen as the ground truth for the rankings of projects. Kendall's tau is calculated for developers and projects separately and is shown in Table 3.

From the results of correlation analysis in Table 3, we have the following observations:

(1) Our proposed BurstBiRank outperforms all baseline methods in identifying Top-20, Top-50, and Top-100 influential developers and projects except for Top-100 developers. This indicates the effectiveness of employing burstiness as the weight between developers and projects for measuring the influence of both developers and projects instead of ignoring edge weight or simply using commit number as edge weight. This result not only agrees with previous study [14] but also conforms to practical intuitions that developers pay more attention on important projects with continuous and regular work on them.

(2) For random walk-based ranking methods (i.e., PageRank and BiRank), higher performance can be obtained with unweighted edges than commit number-weighted edges for identifying high influential (i.e., Top-20) developers and projects, while it is just the opposite for decomposition-based ranking method (i.e., SVDRank and SVDARank). It indicates that how to measure the edge weight is important to model continuous interactions between developers and projects for influence analysis. This is also a key motivation why we model the edge weight as a function of burstiness, and future improvement can be applied on the form of the function.

(3) Almost all baseline methods have negative correlation results which indicate that the ranking orders by these methods are negatively correlated with the ground truth ranking orders, that is, influential developers or projects are usually ranked after less influential ones by these methods, while our proposed method always shows good positive correlation with the ground truth ranking orders, indicating good stability of our method.

4.4.2. SIR Simulation. In this section, the SIR model [31] is adopted to evaluate the performances of our proposed BurstBiRank and the baseline methods by comparing the ability of information spreading of Top-k developers and projects ranked by each method. In the experiment, BurstBiRank is compared with each baseline method separately. Because two comparing ranking methods usually rank a common group of nodes which will show equal effect in SIR simulation, for each pair of comparison, Top-50 developers (projects) ranked by each method are selected and only those developers (projects) not ranked by both methods are set as initial infected nodes. In each iteration t of SIR simulation, infected nodes randomly select one of their susceptible neighbors and infect it with probability $\alpha = 0.5$, and infected nodes recover to removed status with probability β, which is the reciprocal of the average of all node degrees. The accumulative number of infected nodes N_I is recorded for each iteration. Iteration stops when there is no infected nodes. To avoid the randomness of SIR simulation, 10 experiments are conducted for each pair of methods and the results are averaged as the final result. The final results are shown in Figures 4–7, and several significant observations are found:

(1) BurstBiRank outperforms all baseline methods which indicates the effectiveness of burstiness in identifying influential developers and projects. This finding will inspire developers to work continuously and regularly to obtain high influence in open source software community.

(2) The difference of performance between BurstBiRank and PageRank is larger than that between BurstBiRank and BiRank/SVDRank/SVDARank. As we know, PageRank is designed for unipartite network while BiRank, SVDRank, and SVDARank are special ranking methods for bipartite network which distinguish types of nodes and employ the mutual reinforcement between different types of nodes during ranking. This means it is better to

Table 3: Correlation analysis.

Method	Top-20		Top-50		Top-100	
	Project	Developer	Project	Developer	Project	Developer
PageRank-UW	0.1618	−0.0526	0.0211	0.0378	0.1190	0.0244
PageRank-CN	−0.0686	−0.1297	−0.0462	0.0299	0.1232	0.0272
SVDRank-UW	−0.2480	−0.0897	−0.3631	0.0197	0.0105	0.0773
SVDRank-CN	0.0831	0.1221	0.0173	0.0092	0.0428	0.0321
SVDARank-UW	−0.2000	0.0011	−0.3669	0.0200	0.0121	0.0700
SVDARank-CN	0.1024	−0.0263	0.0305	0.0121	0.0503	0.0435
BiRank-UW	0.0390	−0.0263	0.0296	0.0395	0.1195	0.0071
BiRank-CN	−0.0474	−0.2054	0.0207	0.0201	0.0314	0.0534
BurstBiRank	**0.1633**	**0.1604**	**0.0315**	**0.1221**	**0.1257**	**0.0843**

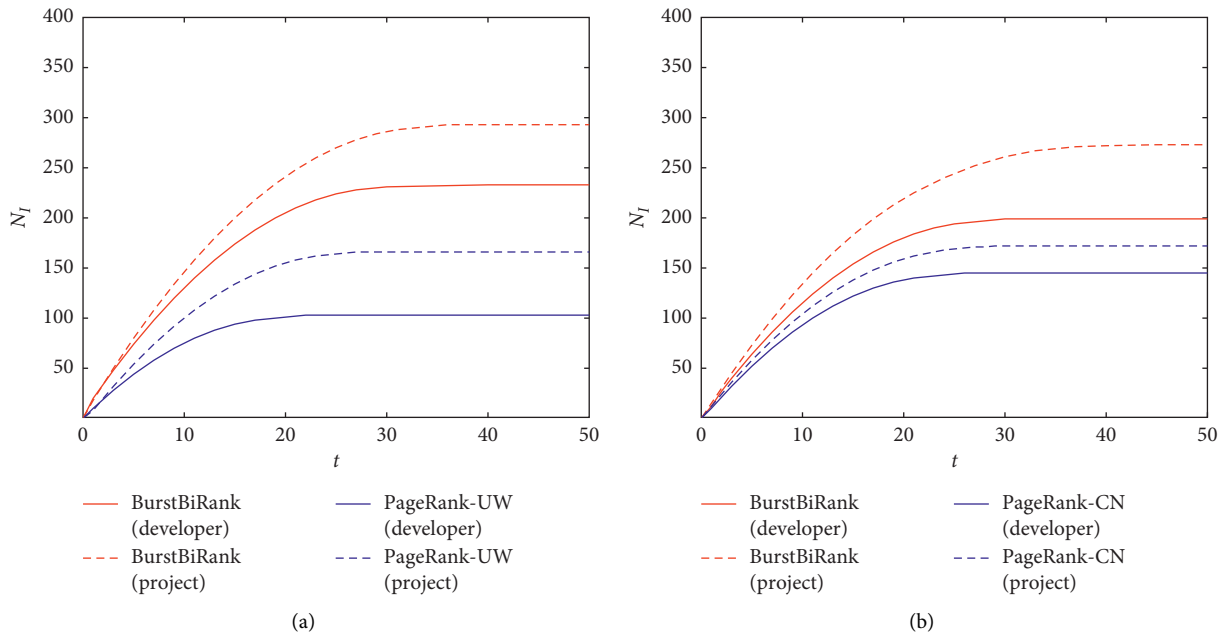


Figure 4: Performance comparison between BurstBiRank and PageRank. (a) BurstBiRank vs. PageRank-UW. (b) BurstBiRank vs. PageRank-CN.

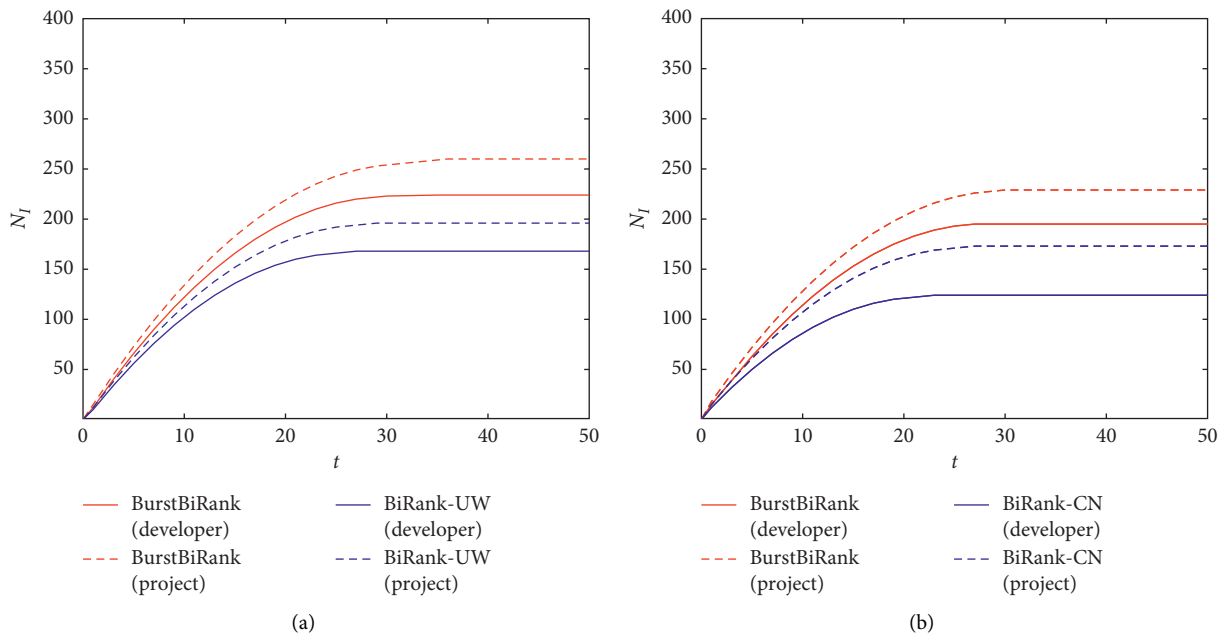


Figure 5: Performance comparison between BurstBiRank and BiRank. (a) BurstBiRank vs. BiRank-UW. (b) BurstBiRank vs. BiRank-CN.

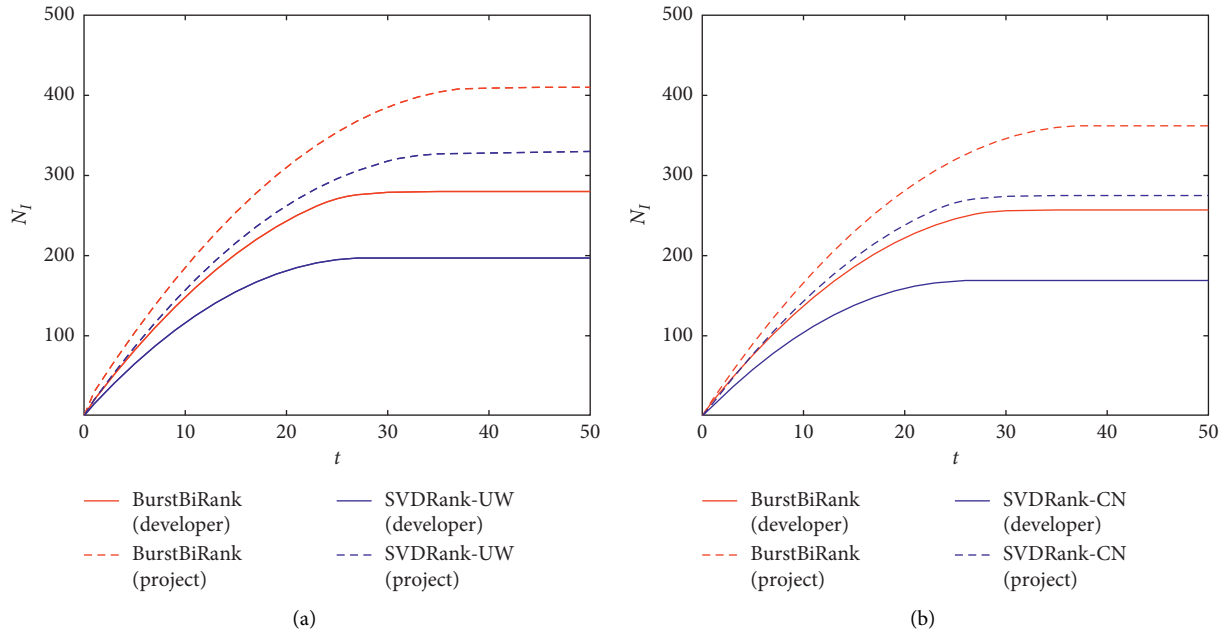


Figure 6: Performance comparison between BurstBiRank and SVDRank. (a) BurstBiRank vs. SVDRank-UW. (b) BurstBiRank vs. SVDRank-CN.

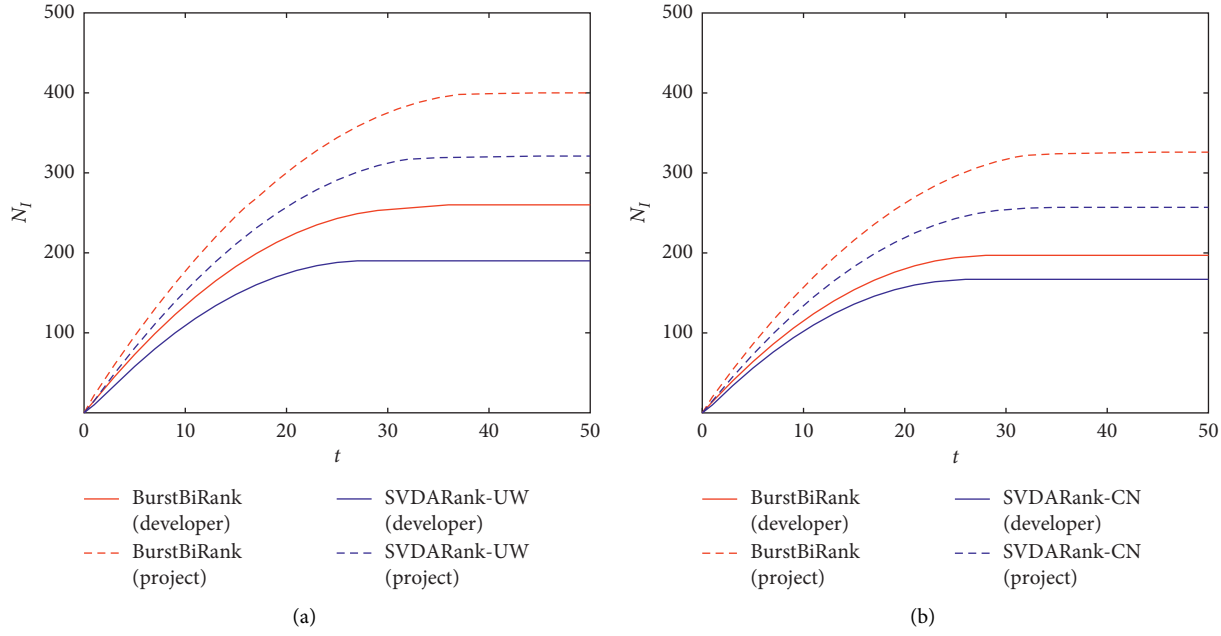


Figure 7: Performance comparison between BurstBiRank and SVDARank. (a) BurstBiRank vs. SVDARank-UW. (b) BurstBiRank vs. SVDARank-CN.

mutually co-rank developers and projects than to simply mix them up.

4.5. Case Study. In addition to correlation analysis and SIR simulation, we further do a detailed case study to show the effectiveness of our model in identifying influential developers and projects. Top-20 developers and projects ranked by BurstBiRank are shown in Tables 4 and 5, respectively, with their rankings in baseline methods.

From Table 4, we can see some major contributors to famous projects can be identified by BurstBiRank but they have lower rankings in baseline methods. For example, Marco Pivetta (GitHub ID: Ocramius), a major contributor of both ZendFramework and Doctrine ORM, is not ranked in Top-20 by BiRank-CN, SVDRank-UW, SVDRank-CN,

Table 4: Top-20 developer ranking.

Developer	Ranking								
	BurstBiRank	PageRank-UW	PageRank-CN	BiRank-UW	BiRank-CN	SVDRank-UW	SVDRank-CN	SVDARank-UW	SVDARank-CN
web-flow	1	1	1	1	1	1	1	1	1
StyleCIBot	2	4	6	4	5	2	16	2	16
QUVAUNKR	3	2	2	2	2	8	3	6	5
translatewiki	4	7	7	7	9	—	11	—	—
dereuromark	5	8	8	8	7	4	8	4	11
invalid-email-	6	3	4	3	3	—	—	20	—
weierophinney	7	6	5	6	4	9	—	—	7
fabpot	8	5	3	5	6	—	12	15	—
GrahamCampbell	9	14	16	14	—	3	7	3	4
s-nakajima	10	16	14	16	11	6	—	8	—
freekmurze	11	—	—	—	—	15	—	—	13
CTYLOQFP	12	17	19	—	—	10	9	—	9
Nyholm	13	—	—	—	—	7	—	9	—
legoktm	14	—	—	20	—	—	—	17	—
yunosh	15	10	9	10	13	—	—	—	20
FaustBrian	16	—	—	—	19	20	18	19	—
FHWXWWSE	17	—	13	—	—	—	20	—	19
Ocramius	18	19	12	18	—	—	—	—	—
taylorotwell	19	—	—	—	20	—	—	—	—
stronk7	20	12	10	15	14	17	—	—	—

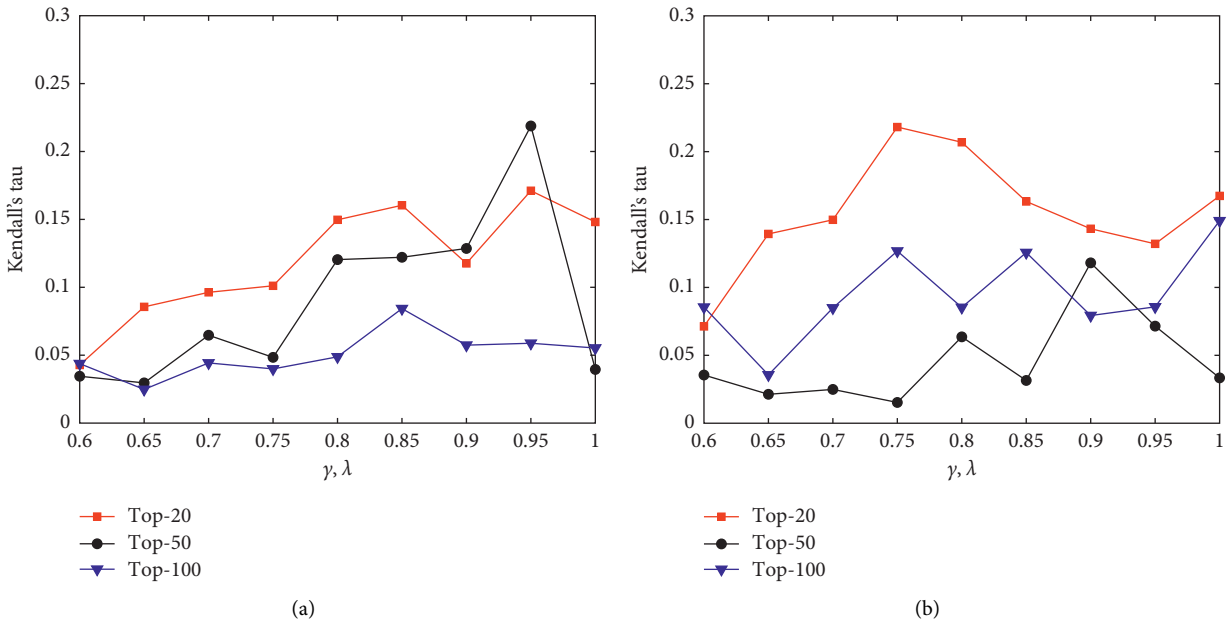


Figure 8: Parameter analysis for (a) developers' ranking performance and (b) projects' ranking performance.

SVDARank-UW, and SVDARank-CN. Taylor Otwell (GitHub ID: taylorotwell), the creator and major contributor to Laravel, is only identified by our BurstBiRank and BiRank-CN.

As for projects, famous PHP projects like Symfony (GitHub ID: symfony/symfony) and MediaWiki (GitHub ID: wikimedia/mediawiki) can also be identified as Top-20 influential projects by our method but with lower rank by baseline methods.

BurstBiRank and the baseline methods all can identify several influential developers and projects, but some less popular developers and projects are also given a high rank because the dataset is a real-world dataset and only a little filtering operations are applied to it. To sum up about correlation analysis, SIR simulation, and case study, our proposed BurstBiRank outperforms baseline methods and can identify some influential developers and projects of real-world open source software community. But further improvement should be conducted.

4.6. Parameter Analysis. In this section, we investigate how the performance varies with the hypermeters that balance the prior beliefs and diffusion scores. For simplicity, we constrain γ to be equal to λ, and Kendall's tau is adopted to indicate the model's performance. Figure 8 shows the raking performance by varying the balance parameters γ and λ from 0.6 to 1. For best ranking performance, the balance

Table 5: Top-20 project ranking.

Project	Ranking								
	BurstBiRank	PageRank-UW	PageRank-CN	BiRank-UW	BiRank-CN	SVDRank-UW	SVDRank-CN	SVDARank-UW	SVDARank-CN
magento/magento2	1	1	1	1	1	15	9	—	6
Wikia/app	2	2	2	2	2	7	3	10	2
brion/mediawiki-	3	3	6	4	8	—	—	—	—
owncloud/core	4	7	4	7	4	1	2	1	4
kaltura/server	5	6	5	6	5	—	6	18	10
oroinc/platform	6	20	19	—	15	13	—	8	—
ZimbraOS/zm-aspel	7	4	8	3	6	—	—	—	—
Automattic/jetpack	8	5	3	5	3	—	13	—	15
wikimedia/mediawi	9	10	12	—	—	19	—	—	—
svick/mediawiki	10	12	17	15	19	—	—	9	8
PrestaShop/PrestaS	11	11	10	9	7	10	11	12	11
orocrm/platform	12	—	—	—	—	—	18	—	—
joomla/joomla-cms	13	15	14	13	13	9	—	7	—
symfony/symfony	14	14	15	14	17	—	—	17	—
orocrm/crm	15	—	—	—	—	20	—	—	—
orocommerce/oroco	16	—	—	—	—	18	16	—	17
ec-europa/platform-	17	19	—	19	20	—	20	20	—
akeneo/pim-	18	—	7	—	9	—	5	—	7
ILIAS-eLearning/IL	19	—	—	—	—	—	—	—	20
cakephp/cakephp	20	—	9	—	10	2	1	3	1

parameters γ and λ are set not equal to 1, indicating the prior beliefs are useful for ranking developers and projects.

5. Conclusions

In this work, we aim at identifying influential developers and projects in open source software community. Continuous interactions between developers and projects are modeled as a burstiness-weighted bipartite network, and an iterative diffusion process is applied on it to calculate ranking scores for developers and projects. The proposed BurstBiRank is evaluated against four baseline methods on a real-world GitHub dataset. Extensive experimental analysis and case study show BurstBiRank outperforms baseline methods in both correlation analysis and SIR simulation.

The basic idea behind BurstBiRank is measuring the tie strength between developers and projects by the burstiness of the continuous interactions between them with an intuitively reasonable assumption that more regular interactions mean stronger ties. Under our framework, burstiness can be employed into the developer-project bipartite network by any linear or nonlinear functions, but in our experiment, a linear function is adopted for simplicity. In addition to burstiness, there are other metrics in human dynamics like memory, which may reflect the tie strength between developers and projects. Attributes of developers and projects such as programming language also affect rankings of them. In future work, we will adopt more types of functions, more metrics in human dynamics, and more attributes of developers and projects in our framework.

Data Availability

The data used in this study can be accessed via https://ghtorrent.org/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant no. 61872002), the University Natural Science Research Project of Anhui Province (grant no. KJ2019A0037), the University Collaborative Innovation Program of Anhui Province (grant no. GXXT- 2019-013), and the Doctoral Scientific Research Foundation of Anhui University (grant no. Y040418194).

References

[1] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, 2018.

[2] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 99, pp. 1–12, 2019.

[3] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative web service quality prediction via exploiting matrix factorization and network map," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126–137, 2016.

[4] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, p. 1, 2019.

[5] S. Wan, Y. Xia, and L. Qi, "Automated colorization of a grayscale image with seed points propagation," *IEEE Transactions on Multimedia*, vol. 22, pp. 1756–1768, 2020.

[6] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2021.

[7] H. Borges, A. Hora, and M. T. Valente, "Predicting the popularity of github repositories," in *Proceedings of the 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, pp. 1–10, Ciudad Real, Spain, 2016.

[8] M. Joblin, S. Apel, C. Hunsen, and W. Mauerer, "Classifying developers into core and peripheral: an empirical study on count and network metrics," in *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pp. 164–174, Buenos Aires, Argentina, 2017.

[9] Q. Xuan, C. Fu, and L. Yu, "Ranking developer candidates by social links," *Advances in Complex Systems*, vol. 17, no. 07n08, Article ID 1550005, 2014.

[10] Y. Hu, S. Wang, Y. Ren, and K. K. R. Choo, "User influence analysis for github developer social networks," *Expert Systems with Applications*, vol. 108, pp. 108–118, 2018.

[11] K. Inoue, R. Yokomori, T. Yamamoto, M. Matsushita, and S. Kusumoto, "Ranking significance of software components based on use relations," *IEEE Transactions on Software Engineering*, vol. 31, no. 3, pp. 213–225, 2005.

[12] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.

[13] X. He, M. Gao, M.-Y. Kan, and D. Wang, "Birank: towards ranking on bipartite graphs," *IEEE Transactions on Knowledge and Data*, vol. 29, no. 1, pp. 57–71, 2016.

[14] D.-C. Yan, Z.-W. Wei, X.-P. Han, and B.-H. Wang, "Empirical analysis on the human dynamics of blogging behavior on github," *Physica A: Statistical Mechanics and Its Applications*, vol. 465, pp. 775–781, 2017.

[15] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol. 650, pp. 1–63, 2016.

[16] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[17] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang, "Dynamics of information diffusion and its applications on complex networks," *Physics Reports*, vol. 651, pp. 1–34, 2016.

[18] Y. Zhao, H. Li, S. Wan et al., "Knowledge-aided convolutional neural network for small organ segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1363–1373, 2019.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," Technical report, Stanford InfoLab, Stanford, CA, USA, 1999.

[20] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[21] T. H. Haveliwala, "Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[22] Y. Liu, B. Gao, T.-Y. Liu et al., "Browserank: letting web users vote for page importance," in *Proceedings of the31st Annual International ACM SIGIR Conference*, pp. 451–458, Singapore, 2008.

[23] L. Lu, Y.-C. Zhang, C. H. Yeung, and T. Zhou, "Leaders in social networks, the delicious case," *PloS One*, vol. 6, no. 6, 2011.

[24] Q. Li, T. Zhou, L. Lü, and D. Chen, "Identifying influential spreaders by weighted leaderrank," *Physica A: Statistical Mechanics and Its Applications*, vol. 404, pp. 47–55, 2014.

[25] S. Xu, P. Wang, and C. Zhang, "Identification of influential spreaders in bipartite networks:A singular value decomposition approach," *Physica A: Statistical Mechanics and Its Applications*, vol. 513, pp. 297–306, 2019.

[26] F. Morone, G. Del Ferraro, and H. A. Makse, "The k-core as a predictor of structural collapse in mutualistic ecosystems," *Nature Physics*, vol. 15, no. 1, pp. 95–102, 2019.

[27] W. Pan, H. Ming, C. Chang, Z. Yang, and D.-K. Kim, "Elementrank: ranking java software classes and packages using a multilayer complex network-based approach," *IEEE Transactions on Software Engineering*, 2019.

[28] K.-I. Goh and A.-L. Barabási, "Burstiness and memory in complex systems," *EPL (Europhysics Letters)*, vol. 81, no. 4, p. 48002, 2008.

[29] G. Gousios, "The ghtorrent dataset and tool suite," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, pp. 233–236, San Francisco, CA, USA, 2013.

[30] W. R. Knight, "A computer method for calculating kendall's tau with ungrouped data," *Journal of the American Statistical Association*, vol. 61, no. 314, pp. 436–439, 1966.

[31] R. Yang, B.-H. Wang, J. Ren et al., "Epidemic spreading on heterogeneous networks with identical infectivity," *Phys. Lett. A*, vol. 364, no. 3-4, pp. 189–193, 2007.

[32] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of Modern Physics*, vol. 87, no. 3, p. 925, 2015.

Hindawi
Complexity
Volume 2020, Article ID 8810759, 15 pages
https://doi.org/10.1155/2020/8810759

Research Article

Improving the Performance of Whale Optimization Algorithm through OpenCL-Based FPGA Accelerator

Qiangqiang Jiang,[1] Yuanjun Guo,[1] Zhile Yang,[1] Zheng Wang,[1] Dongsheng Yang,[2] and Xianyu Zhou[2]

[1]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China*
[2]*Intelligent Electrical Science and Technology Research Institute, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Yuanjun Guo; yj.guo@siat.ac.cn

Received 25 August 2020; Revised 23 November 2020; Accepted 3 December 2020; Published 16 December 2020

Academic Editor: Shangce Gao

Whale optimization algorithm (WOA), known as a novel nature-inspired swarm optimization algorithm, demonstrates superiority in handling global continuous optimization problems. However, its performance deteriorates when applied to large-scale complex problems due to rapidly increasing execution time required for huge computational tasks. Based on interactions within the population, WOA is naturally amenable to parallelism, prompting an effective approach to mitigate the drawbacks of sequential WOA. In this paper, field programmable gate array (FPGA) is used as an accelerator, of which the high-level synthesis utilizes open computing language (OpenCL) as a general programming paradigm for heterogeneous System-on-Chip. With above platform, a novel parallel framework of WOA named PWOA is presented. The proposed framework comprises two feasible parallel models called partial parallel and all-FPGA parallel, respectively. Experiments are conducted by performing WOA on CPU and PWOA on OpenCL-based FPGA heterogeneous platform, to solve ten well-known benchmark functions. Meanwhile, other two classic algorithms including particle swarm optimization (PSO) and competitive swarm optimizer (CSO) are adopted for comparison. Numerical results show that the proposed approach achieves a promising computational performance coupled with efficient optimization on relatively large-scale complex problems.

1. Introduction

Swarm optimization or evolutionary algorithms have demonstrated their significance in a wide range of scientific and practical problems [1–5]. Recent years, more and more research studies' focus has been on multiobjective problems and artificial intelligence [6–9]. Whale optimization algorithm (WOA), a novel swarm intelligence-based metaheuristic algorithm, was proposed by Mirjalili and Lewis in 2016 [10]. Inspired by the special hunting behavior of humpback whales, WOA shows better performance compared with several existing popular methods and has drawn great research attention. Typically, Abdel-Basset et al. [11] integrated WOA with a locals search strategy for tackling the permutation flow shop scheduling problem. Mafarja and Mirjalili [12] proposed a hybrid WOA with simulated annealing for feature extraction. Aljarah et al. [13] introduced WOA-based trainer to train multilayer perceptron (MLP) neural networks. Moreover, there are also research bodies trying to tackle other diverse problems using WOA, such as multiobjective optimization [14–16], image processing [17–19], software testing [20], and power system applications [21, 22].

However, large-scale, multiple constraints and complex scenarios usually appear in actual engineering optimization problems, such as job shop scheduling, mixed unit commitment problem, and automatic path planning. Furthermore, high requirements in response speed and real-time performance need to be satisfied when solving problems above. In this situation, most optimization algorithms including WOA might get stuck in the executing dilemma. As the scale and complexity of the problem increase, the

execution time of WOA will increase rapidly, which leads to time-performance deterioration [23]. With inherent parallelism of WOA, the abovementioned problem can be tackled by applying parallel algorithm developed targeting specific accelerating platforms. In recent years, experts and scholars have tried to implement various swarm optimization algorithms employing state-of-the-art technologies such as multicore (message passing interface-MPI, OpenMP), distributed (MapReduce, Spark), and heterogeneous computing-based parallel platforms (graphics processing unit-GPU, FPGA).

Heterogeneous computing refers to using dedicated hardware devices with different architectures to execute time-consuming tasks, balancing the computational load of CPU. GPU is a classical parallel computing device, widely used in graphics visualization, image/video processing, scientific computing, deep learning, and so on. Nevertheless, with the increasing deployment of GPU, energy consumption and heat dissipation have become severe limitations for system extension, as well as brings heavy environmental pressure to human society [24]. In light of this, some researchers begin to choose other hardware devices to alleviate pressure caused by GPU. FPGA, a novel parallel accelerator, possesses powerful parallel computing capability and flexible programmability, while maintaining the advantage of low power consumption [25]. Traditional FPGA design, however, has drawbacks of high development difficulty and time consumption. Recently, Intel provides a development kit for software users, making it possible to deploy OpenCL program on FPGA. Consequently, developers can rapidly implement FPGA-based heterogeneous applications through OpenCL API thus reducing the development cost and time-to-market.

This research proposes high-performance parallel WOA (PWOA), with implementation on FPGA to effectively solve large-scale and complex optimization problems. More specifically, the main contributions of this paper are presented as follows:

(1) A novel heterogeneous parallel framework of WOA based on OpenCL-based FPGA accelerator.

(2) Two efficient models including partial parallel model and all-FPGA parallel model, with program flow design and dataflow analysis.

(3) Several diverse numerical experiments are conducted with ten selected benchmark functions. By comparing with sequential WOA executing on CPU, the proposed PWOA based on two parallel models achieves higher execution performance.

The rest of this paper is organized as follows: Section 2 represents a substantial literature review on exploration for parallel optimization algorithms. The theory of WOA and OpenCL-based FPGA heterogeneous accelerating platform is introduced in Section 3. Section 4 describes FPGA implementation of the proposed PWOAs with two parallel models and followed by the experimental results and statistical analysis in Section 5. Finally, conclusions are given in Section 6.

2. Related Work

Swarm optimization algorithms including WOA encounter challenges that the optimization performance decreases due to extensive computational cost when solving problems with high-dimension and complex mathematical model. To overcome these challenges, researchers have designed parallel swarm algorithms with implementation on various platforms. In recent years, distributed and parallel particle swarm optimization (PSO) has been implemented. Some studies [26–30] applied GPU to parallelize PSO, putting forward diverse parallel strategies. Hajewski and Oliveira [31] developed fast cache-aware parallel PSO relying on OpenMP. Ant colony optimization (ACO) [32] and artificial bee colony (ABC) [33] were also parallelized by GPU. Concerning brain storm optimization (BSO), Jin and Qin [34] presented GPU-based manner whilst Ma et al. [35] proposed parallelized BSO algorithm based on Spark framework for association rule mining. Similar works in [36, 37] used GPU and FPGA to accelerate genetic algorithm (GA). What deserves attention is that Garcia et al. [38] achieved parallel implementation and comparison of teaching-learning based optimization (TLBO) and Jaya on many-core GPU. As for WOA, Khalil et al. [39] proposed a simple and robust distributed WOA using Hadoop MapReduce, reaching a promising speedup.

It can be concluded that there are several typical kinds of parallel techniques including OpenMP, MapReduce, Spark, and heterogeneous architecture based on dedicated accelerators, such as GPU and FPGA. GPU becomes popular for general-purpose parallel computing as developing parallel swarm intelligence algorithms via GPU has successfully gained remarkable performance improvement [40]. Recently, FPGA is gradually applied to heterogeneous computing and algorithms accelerating based on OpenCL, which benefits from its high-parallelism, better energy efficiency, and flexible programmability [41–44]. The experiments conducted by [45] showed that swarm algorithms on FPGAs achieved a better speedup than that on GPUs and multicore CPUs. Nevertheless, designing a near-optimal accelerator is not an easy task. Implementing CPU-oriented codes on FPGA rarely increases the performance and even reduces the performance compared to CPU. Therefore, it requires not only the digital design expertise but also software skills to form appropriate OpenCL codes [46].

Few research works have been investigated on FPGA implementation of swarm optimization algorithms, especially WOA. Our prior work [47] explored WOA based on partial parallel scheme and deployed it on the FPGA heterogeneous platform. Then, empirical results using classic benchmarks proved the consequential advance of the

proposed methology in execution performance and convergence speed. In this paper, motivated by the previous studies, a novel PWOA scenario with two parallel models encompassed is further exploited via FPGA. Meanwhile, more diverse benchmarks are used to verify the effectiveness of PWOA based on FPGA parallel framework and its computational performance for large-scale complex problems.

3. Whale Optimization Algorithm and Acceleration Platform

3.1. Basic WOA Algorithm. The WOA algorithm constitutes two main phases, exploitation and exploration, through emulating shrinking encircling, bubble-net attacking, and searching for preys. The following subsections explain in detail the mathematical models of each phase.

3.1.1. Exploitation Phase (Encircling and Bubble-Net Attacking). To hunt preys, humpback whales first recognize the location of preys and encircle them. The mathematical model of shrinking encircling is represented by the following equations:

$$\mathbf{D} = \left| C \cdot \mathbf{X}^*_{(t)} - \mathbf{X}_{(t)} \right|, \tag{1}$$

$$\mathbf{X}_{(t+1)} = \mathbf{X}^*_{(t)} - A \cdot \mathbf{D}, \tag{2}$$

where $\mathbf{X}$ is the position vector, $\mathbf{X}^*$ represents the position of the best solution obtained so far, t indicates the current number of iteration, $||$ denotes the absolute operation, and $\cdot$ means an element-by-element multiplication. A and C are two parameters, which are calculated as follows:

$$A = 2a \cdot r - a, \tag{3}$$

$$C = 2 \cdot r, \tag{4}$$

where a is linearly decreasing from 2 to 0 through iterations (in both exploitation and exploration phases) and r is a random number in [0, 1]. The value of a is calculated by $a = 2 - t\,(2/\text{MaxIter})$, and MaxIter is the maximum number of iterations.

Another method used in the exploitation phase is spiral updating position, which in coordination with aforementioned shrinking encircling constitutes the bubble-net attacking strategy of humpback whales. The mathematical equations are as follows:

$$D' = \left| \mathbf{X}^*_{(t)} - \mathbf{X}_{(t)} \right|, \tag{5}$$

$$\mathbf{X}_{(t+1)} = D' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{X}^*_{(t)}, \tag{6}$$

where b is a constant for determining the shape of the logarithmic spiral and l is a random number in $[-1, 1]$. Shrinking encircling and spiral updating position are used simultaneously during exploitation phase. The mathematical model is as follows:

$$\mathbf{X}_{(t+1)} = \begin{cases} \mathbf{X}^*_{(t)} - A \cdot \mathbf{D}, & p < 0.5, \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{X}^*_{(t)}, & p \geq 0.5, \end{cases} \tag{7}$$

where p is a random value in $[-1, 1]$ which stands for a probability of 50% to choose either the shrinking encircling method or the spiral-shaped mechanism to update the position of whales during optimization process.

3.1.2. Exploration Phase (Searching for Preys). In addition to exploitation phase, a stochastic searching technique is also adopted to enhance the exploration in WOA. Unlike exploitation, a random whale $\mathbf{X}_{\text{rand}}$ is selected from swarm to navigate the search space, so as to find a better optimal solution (prey) than the existing one. This phase can efficiently prevent the algorithm from falling into local optima stagnation. Subsequently, based on the parameter A, a decision is made on which mechanism to be used for updating the position of whales. Exploration is done if $|A| \geq 1$, meanwhile if $|A| < 1$. The optimization process is mathematically described as follows:

$$\mathbf{D} = \left| C \cdot \mathbf{X}_{\text{rand}} - \mathbf{X}_{(t)} \right|, \tag{8}$$

$$\mathbf{X}_{(t+1)} = \mathbf{X}_{\text{rand}} - A \cdot \mathbf{D}, \tag{9}$$

where $\mathbf{X}_{\text{rand}}$ is a random position of the whale chosen from the current population and C is calculated by equation (4).

Algorithm 1 presents the pseudocode of WOA. At the beginning of the algorithm, an initial random population is generated, and each individual gets evaluated by fitness function and X^* is the current best solution. Then, the algorithm is repeatedly executed until the stop condition is satisfied. At each iteration, search agents update their position according to either a random chosen individual when $|A| \geq 1$, or the optimal solution obtained so far when $|A| < 1$. Depending on p, the WOA algorithm decides on whether using circular or spiral movement.

3.2. OpenCL-Based FPGA Heterogeneous Computing Platform

3.2.1. OpenCL and FPGA. OpenCL, maintained by Khronos Group, is an open standard for general-purpose parallel computing [48]. Various hardware devices, such as CPU, FPGA, GPU, and DSP, are supported for implementing highly efficient and parallel algorithms across heterogeneous computing platform. Additionally, OpenCL specifies a *C*99-based programming API for convenience of software developers. A typical OpenCL program consists of host and kernel sections.

FPGA is a configurable integrated circuit that can be repeatedly reconfigured to perform a huge number of logic functions. It generally includes programmable core logics, hierarchical reconfigurable interconnects, I/O elements, memory blocks, and DSPs. With these substantial logical resources, FPGA achieves an increased programming flexibility compared to application-specific integrated circuits

```
Generate initial population X_i (i = 1, 2, ..., n)
Evaluate the fitness of each search agent
X* = the best search agent
while (t < MaxIter) do
  for each search agent do
    Update a, A, C, l and p
    if (p < 0.5) then
      if (|A| < 1) then
        Update the position of the current search agent by equation (2)
      else if (|A| ≥ 1) then
        Select a random search agent (X_rand)
        Update the position of the current search agent by equation (9)
      end if
    else if (p ≥ 0.5) then
      Update the position of the current search agent by equation (6)
    end if
  end for
  Amend search agents which go beyond the search space
  Calculate the fitness of each search agent
  Replace X* with a better solution (if found)
  t = t + 1
end while
return X*
```

Algorithm 1:

(ASICs). However, traditional development flow on FPGA heavily relies on register transfer level (RTL) descriptions such as Verilog and very high speed integrated circuit hardware description language (VHDL), which incurs high development and verification cost. To address this problem, FPGA vendors such as Intel and Xilinx released OpenCL-based development flow which eases software developers to design FPGA-based applications, making this process more efficient.

3.2.2. Intel FPGA SDK for OpenCL. The Intel FPGA SDK for OpenCL [49] entitles developers to create high-level FPGA implementation with OpenCL. This SDK generates a heterogeneous computing environment where OpenCL kernels are compiled by Altera Offline Compiler (AOC) for programming FPGA at runtime. In this paradigm, Intel achieves design optimization while hiding low-level hardware details of FPGA. Subsequently, FPGA has gradually been applied to a wide range of fields such as image and video processing [42, 50], deep learning [51–53], and intelligent optimization algorithm [46].

OpenCL-based FPGA logic framework is illustrated in Figure 1 where several modules are specifically explained as follows:

(1)Kernel pipeline: the core module of entire framework, which is an implementation of specific functions. The kernel code is compiled by AOC offline compiler and will be synthesized into highly parallel optimized logic circuit referring to the internal architecture of FPGA.

(2)Processor: a host processor, typically CPU, used to control programs running on FPGA device.

(3)DDR: off-chip memory, including global and constant memory in the OpenCL memory model. Intel Cyclone V FPGA device used in this context has a DDR3 with a capacity of 1 GB. By default, the constant cache size is 16 KB and can be modified in accordance with practical requirements.

(4)PCI-e: high-speed data exchanging interface, responsible for transporting data and instruction between host and device.

(5)On-chip memory: internal memory of FPGA device, equivalent to local and private memory in the OpenCL memory model. With small capacity but high speed, it is mainly used for storing input and output temporary data, reducing the number of accesses to global memory. Thus, we may take advantage of on-chip memory to improve the efficiency of OpenCL program.

(6)Local memory interconnect: a bridge between executing unit and memory.

(7)External memory controller and PHY: a controller which is in charge of controlling data sending and receiving via DDR.

4. Parallel Whale Optimization Algorithm Based on FPGA

With the descriptions and definitions above, the framework of WOA can be summarized as shown in the left flowchart in Figure 2. Note, meanwhile, that the right flowchart is a simplified framework of WOA which is mainly composed of initialization, swarm updating, fitness calculation, and swarm evaluation. Similar to other swarm optimization

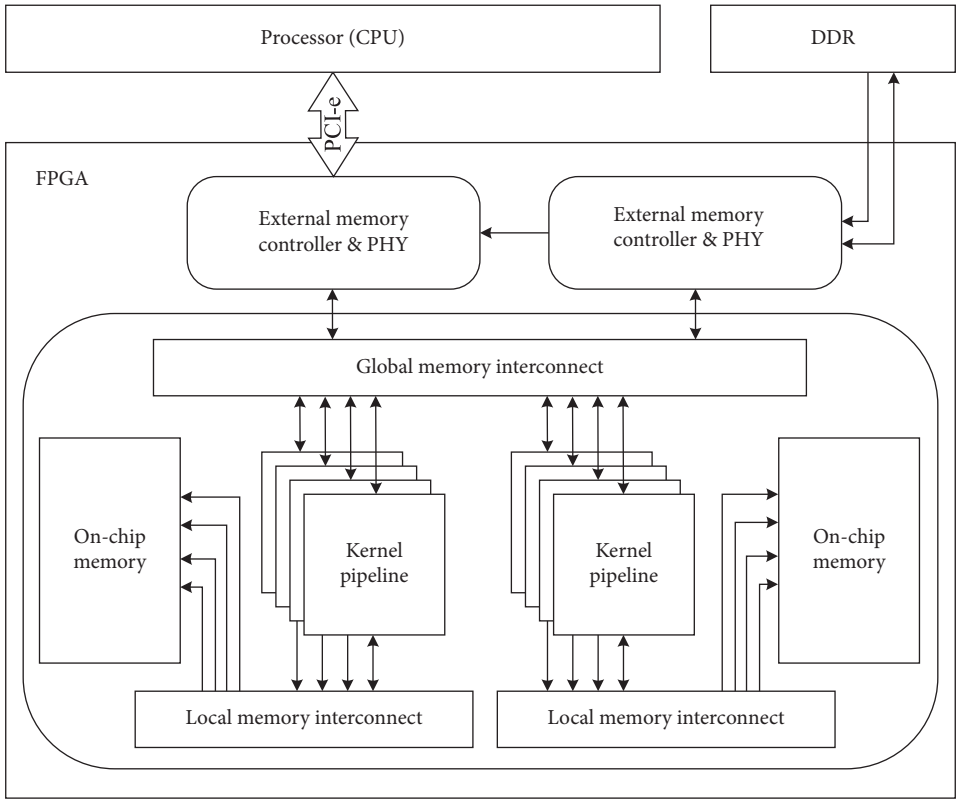


Figure 1: Architecture of heterogeneous platform with CPU and FPGA.

algorithms, WOA unavoidably suffers from this drawback of time-consuming operations such as updating swarm and calculating fitness, which greatly limits its execution speed [45]. Thanks to natural parallelism, the components utilized to implement swarm updating and fitness calculation in WOA can be executed concurrently. Within swarm updating phase, the positions of searching whales are updated separately by corresponding moving mechanism, more biologically simulating a real hunting process. For the remaining two phases, the initialization keeps primary ideology in this work for it has little effect on computational performance, whereas the evaluation is synchronous and cannot be paralleled.

This section will propose parallel WOA based on FPGA heterogeneous computing platform. In order to reach efficient acceleration, some compute-intensive tasks of WOA need to be transferred to FPGA side for parallel execution whilst CPU performs the remaining tasks. The parallel model can be divided into partial parallel and all-FPGA parallel by assigning different tasks to CPU and FPGA. Below is the PWOA implementation, which is described in two aspects: program flow design and dataflow analysis.

4.1. Initialization. Initialization mainly prepares basic data needed during the whole phase of WOA, including generating random numbers and initial population. This process is carried out at the beginning stage of WOA and executed only once. On top of this, C/C++ dedicated library for random number generation is applied as OpenCL does not support native random number generator. In this paper, a generic methodology, putting computational task of initialization at CPU side, is adopted into these two proposed parallel models, so as to take full use of computational horsepower from CPU.

Random number generation is a crucial component for WOA. On the one hand, the initial population is composed of whales with a random position. What needs to be ensured is that the value of the random position must be within the range of decision variable according to specific objective functions. On the other hand, there are several random numbers used as coefficients (a, A, C, l, and p) for updating the position of whale, which plays a significant role in optimization performance. Besides, these coefficients appear in each iteration, meaning that data transportation between FPGA and CPU also appears in each iteration. It will become a bottleneck for the running speed of PWOA due to frequent data transportation between FPGA and CPU. To alleviate this drawback, all random numbers required as well as the initial population are generated at CPU side and then sent to FPGA side once via OpenCL global memory. The proposed approach can substantially reduce the time overhead of PWOA.

4.2. Partial Parallel Model-Based PWOA

4.2.1. Program Flow Design. The partial parallel model executes several algorithmic sections in parallel involving the so-called master-slaves model. The partial parallel model-based PWOA (PWOA-PPM) on FPGA is presented in Figure 3.

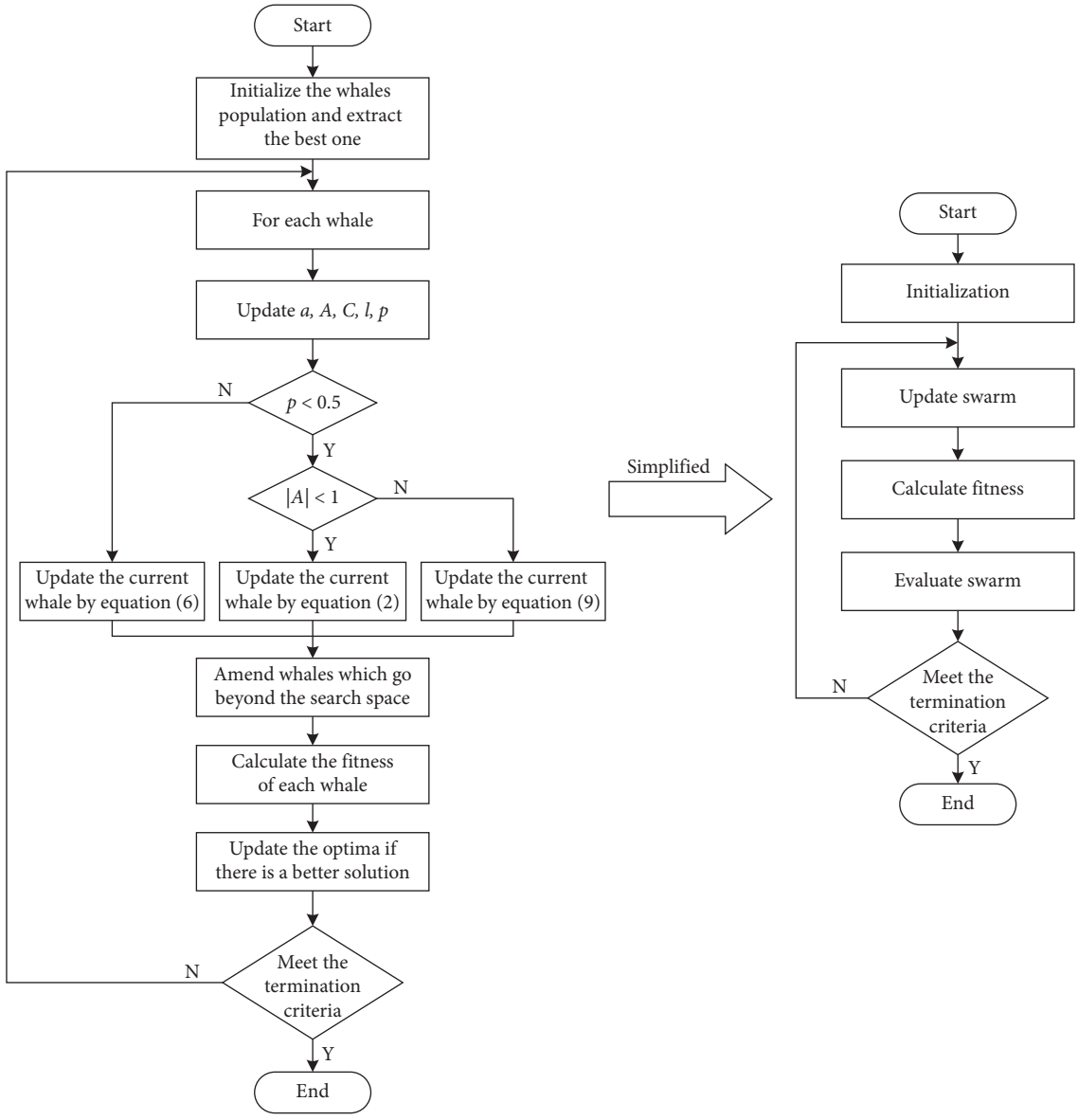


Figure 2: Framework of the WOA algorithm.

Host Program Flow. In terms of host program running on CPU, it undertakes the initialization of PWOA and transfers basic data related to kernel side via OpenCL global memory. Due to the restriction of synchronization, swarm evaluation is put on CPU for sequential executing in this model. After that, host program maintains the basic framework of WOA where it allocates tasks to FPGA, reads computation results from FPGA, and evaluates swarm in each iteration. The evaluation result is also sent to FPGA when host program enqueues task commands which drive kernel function to be executed on FPGA. Such task allocation can make better use of the processing power of CPU but correspondingly cause supernumerary communication overhead between CPU and FPGA.

Kernel Program Flow. FPGA device is used to deploy kernel program and accelerate it. Host offloads computationally intensive tasks onto FPGA for parallel computing. Based on the OpenCL programming model, the parallel parts of the algorithm are mapped to kernel function to be executed by threads (or work items) independently [40, 45]. In the proposed model, a fine-grained strategy is adopted, where each thread takes charge of an individual, calculating fitness and updating position. According to the coefficients (A and p), each thread (individual) performs different mechanisms simultaneously: shrinking encircling, spiral updating, or stochastic searching. Once kernel program finishes executing, the final results are written back to global memory.

4.2.2. Dataflow Analysis between Host and Kernel. In the proposed implementation, the dataflow between host and kernel mostly depends on global memory bandwidth. At host side, the memory buffers are created and the data used are mapped to these buffers, which will be further sent to the

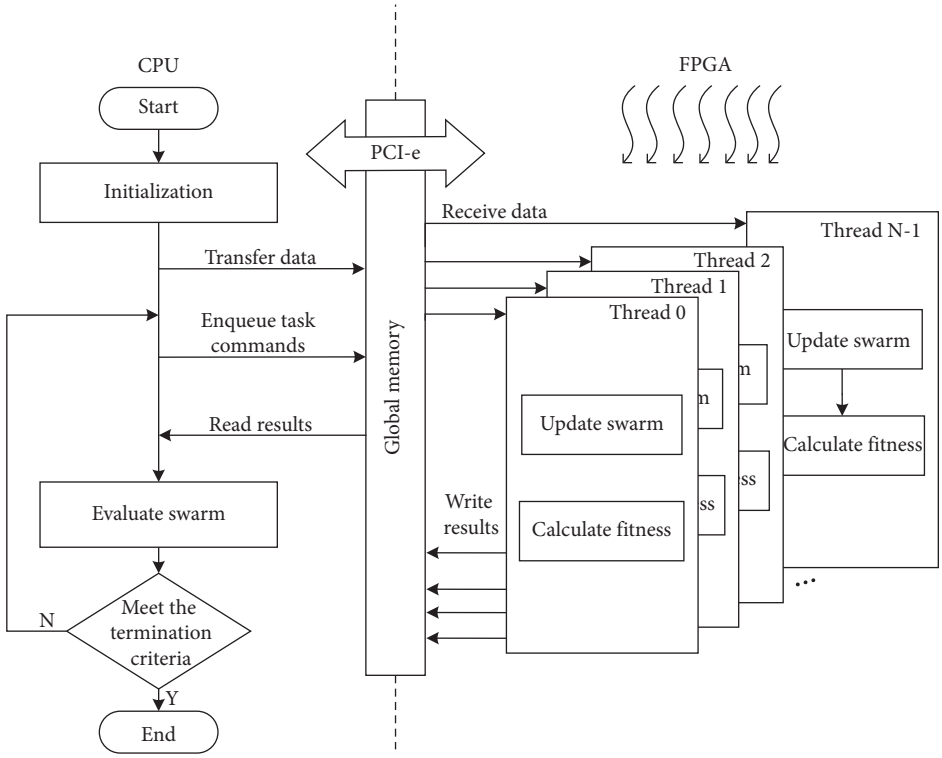


Figure 3: Framework of the partial parallel model-based PWOA (PWOA-PPM).

global memory of kernel via PCI-e. At kernel side, each thread works as a basic processing element, reads data from global memory and complete kernel function. As illustrated in Figure 4, the data set contains positions and fitnesses of all search agents, the global optima X^*, and coefficients (a, A, C, l and p). One block in the "positions" memory block and "fitnesses" memory block represents multi-dimension position information and fitness value of one whale individual, respectively. In the "coefficients" memory block, all coefficients required for a whale during the whole iterations are stored in one block, whereas the "optima" memory block just holds the position of the global best whale.

4.3. All-FPGA Parallel Model Based PWOA

4.3.1. Program Flow Design. In the all-FPGA parallel model, most constituent parts of WOA, except for initialization, are ported to FPGA. The All-FPGA parallel model-based PWOA (PWOA-AFPM) is designed as shown in Figure 5.

Host Program Flow. At host side (CPU), similar to the previous partial parallel model, host program undertakes the initialization of WOA and the basic data related are off-loaded onto kernel side via OpenCL global memory. However, it no longer controls the basic framework of WOA in this parallel model, making a relatively low workload for CPU while a greater computation overhead for FPGA. After completing the above two operations, the host program enqueues task commands to start the kernel program of FPGA and finally reads results from global memory. A dramatic advantage of this design, in comparison with partial parallel model, is minimal communication overhead between CPU and FPGA.

Kernel Program Flow. Within this model, kernel program running on FPGA becomes more complex than the previous model. In addition to receiving data and writing results back to global memory, the evolutionary framework, which contains swarm updating, fitness calculation, and swarm evaluation, is controlled by the kernel. Similarly, the fine-grained model is also applied to make multiple threads executing kernel function in parallel. Nevertheless, cares should be taken when evaluating swarm because all threads share one global optimal solution. To ensure the accuracy of the algorithm, we define memory consistency across threads with respect to memory fences [54, 55]. As depicted in Figure 5, the process in red dotted line is performed as synchronizer, where not only do all threads reach a synchronized state before this process, but also using a better solution obtained by any thread to substitute for the global optimal solution is an atomic operation as well. In this way, all threads can be executed in order, therefore guarantees the evaluation results.

4.3.2. Dataflow Analysis between Host and Kernel. In this model, the dataflow between host and kernel involves global memory and on-chip memory (local memory), which is presented in Figure 6. The same as the previous model, positions and coefficients are transmitted via global memory. To transmit the final result from the kernel to host, it also requires global memory to store this variable. Therefore,

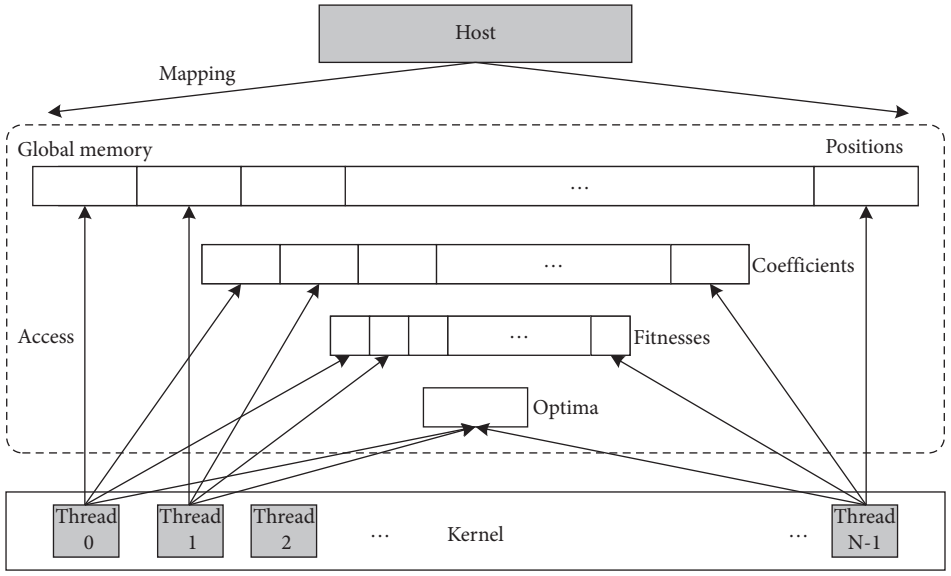


Figure 4: Dataflow of PWOA-PPM between host and kernel.

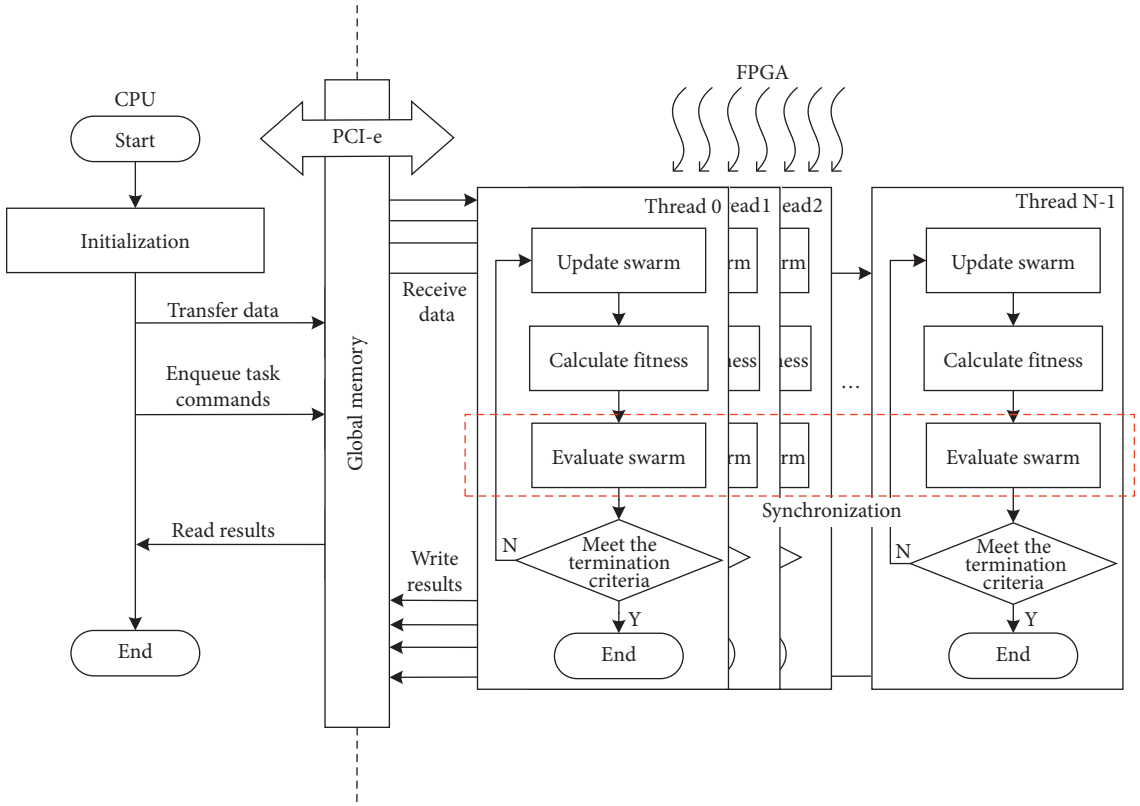


Figure 5: Framework of the All-FPGA parallel model-based PWOA (PWOA-AFPM).

a memory buffer is created by host program at the beginning, requesting a global memory space for the global optima X^*. Usage of on-chip memory from FPGA is a noticeable variation of dataflow between this model and the prior model illustrated in Figure 4. This is because most operations of WOA are executed by FPGA, and it is a rational strategy to utilize on-chip memory comprised of local memory and private memory. Besides, this kind of memory can be directly and efficiently requested during the process of executing kernel. Thus, the intermediate results, such as optima and the fitness of all individuals, are stored into local memory. Furthermore, a more efficient synchronous evaluation process also benefits from the dataset in local memory.

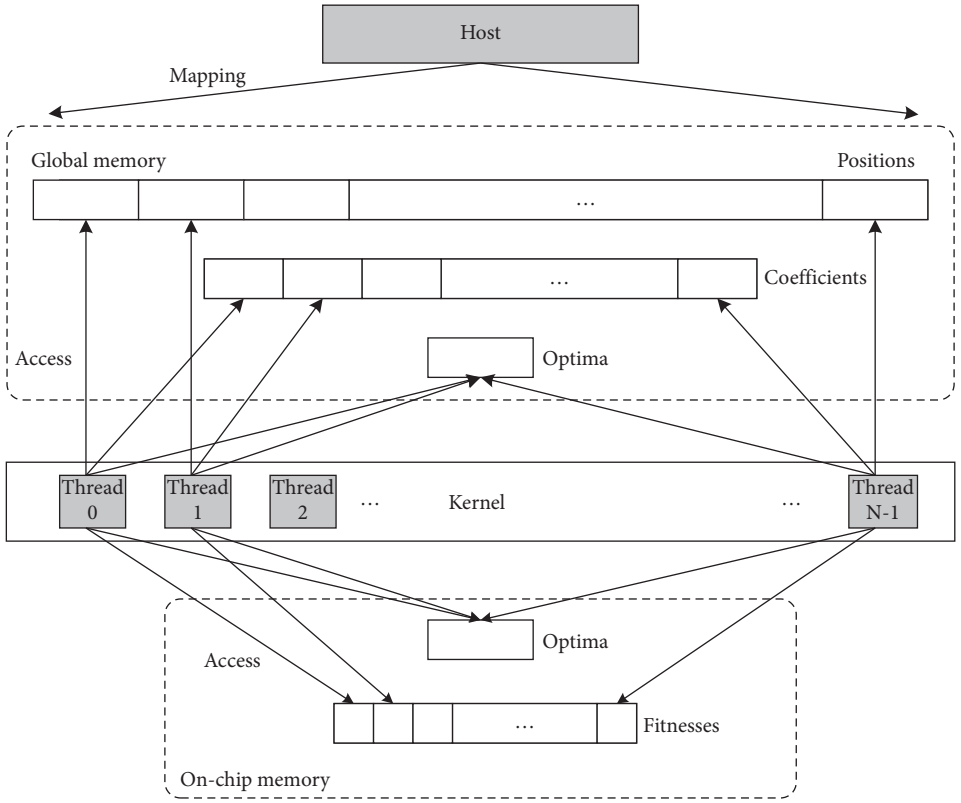


Figure 6: Dataflow of PWOA-AFPM between host and kernel.

5. Numerical Experiment and Analysis

5.1. Experimental Setup. The experimental platform contains two main hardware devices: CPU and FPGA. For CPU platform, Intel Core i5-8250 CPU with 16 GB RAM is used, while for FPGA platform, Intel FPGA Cyclone VGT with 1 GB DDR3 and 64 MB SDRAM is used. The entire development environment is based on Ubuntu 14.04 LTS and Intel FPGA SDK for OpenCL 17.1 version.

In this paper, ten general benchmark functions [56], listed in Table 1, are used to make performance comparisons between serial WOA (CPU implementation) and two parallel models based PWOA (FPGA implementation). Among these benchmark functions, $f1 \sim f5$ are unimodal functions while $f6 \sim f10$ are multimodal functions.

Concerning other parameters in the canonical WOA algorithm, coefficient b in the spiral-updating model is held constant during the whole evaluation process and set to be 1.0. Dimensions including 64 D, 128 D, 256 D, and 512 D are set for the optimization test, and the population size of WOA is dynamically set to be twice the size of the dimension. To verify the performance of the proposed PWOAs, other two canonical algorithms, PSO [57] and competitive swarm optimizer (CSO) [58], are selected for comparison. Additionally, for each implementation with a specific dimension setting, 30 independent runs are executed and the average performance is considered. For each independent run, the maximum number of fitness evaluations (FEs) is set to $1000 \times D$, where D is the search dimension of the test functions.

5.2. Optimization Result and Running Time on Benchmark Functions. By using three WOAs with different schemes and two state-of-the-art algorithms to optimize 10 benchmark functions, experimental data can be obtained as listed in Table 2–5, where *mean* and *time* refer to the average values of optimization result and running time for 30 runs.

Based on numerical values given in above tables, it can be noticed that WOA and PWOAs constructed by two parallel models (PWOA-PPM and PWOA-AFPM) present higher problem solving efficacy than CSO and PSO when optimizing all benchmark functions with several dimensions. As for mean results of all the 10 test cases, WOA and the proposed PWOAs obtain more accurate values, compared with other two algorithms. When optimizing $f1$, $f2$, $f4$, $f8$, and $f10$, the results of WOA and PWOAs maintain a tiny gap with optimal values (0). The proposed algorithms, particularly, can converge to a theoretical optimal value ($f_{\min} = 0$) for $f7$ and $f9$ at any scale. CSO can get more reliable solutions for $f1$, $f2$, $f8$, and $f9$, which, however, are still lower than the proposed algorithms in accuracy. Relatively speaking, PSO hardly converges to an accurate value for most benchmarks. The comparison between WOA and PWOAs shows that the proposed parallel framework based on FPGA heterogeneous platform for WOA maintains intrinsic outstanding global convergence. On top of that, with the increasing of both dimension and population size, the performance of the proposed algorithms are improved for most benchmark functions except for $f3$, $f5$, and $f8$, which indicates that dimension setting affects optimization performance to some

Table 1: Benchmark functions.

Func.	Expression	Range	$f_{\min}$
$f1$	$f(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
$f2$	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$	0
$f3$	$f(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	0
$f4$	$f(x) = \max_{1 \le i \le D} \lvert x_i \rvert$	$[-100, 100]^D$	0
$f5$	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	0
$f6$	$f(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]^D$	$-418.983 \times D$
$f7$	$f(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	0
$f8$	$f(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D} - e^{D^{-1}\sum_{i=1}^{D}\cos(2\pi x_i)}}} + 20 + e$	$[-32, 32]^D$	0
$f9$	$f(x) = \sum_{i=1}^{D} (x_i^2/4000) - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^D$	0
$f10$	$f(x) = (\pi/D)\{10\sin(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$, $y_i = 1 + (x_i + 1/4)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$	$[-50, 50]^D$	0

Table 2: Comparison between the proposed algorithms and the state-of-the-art algorithms (64D).

Func.	PWOA-PPM		PWOA-AFPM		WOA		CSO		PSO	
	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)
$f1$	**2.32E − 105**	0.1625	2.73E − 104	**0.1009**	1.98E − 103	0.2905	9.28E − 07	0.1715	7.52E − 07	0.2689
$f2$	**3.71E − 62**	0.1621	2.69E − 61	**0.0868**	6.50E − 61	0.3027	5.57E − 04	0.1746	8.98E + 01	0.2741
$f3$	6.36E + 02	0.1612	**3.47E + 02**	**0.1028**	2.67E + 03	0.2859	8.75E + 03	0.1711	1.76E + 04	0.2513
$f4$	9.16E − 16	0.1589	1.04E − 16	**0.9639**	**6.91E − 17**	0.2903	1.50E + 01	0.1598	3.42E + 01	0.2842
$f5$	2.73E − 01	0.1747	2.53E − 01	**0.1054**	**1.72E − 01**	0.2918	1.70E + 02	0.1803	3.22E + 02	0.2892
$f6$	**−2.65E + 04**	0.2068	−2.63E + 04	**0.1223**	−4.14E + 03	0.6707	−2.19E + 04	0.2945	−1.78E + 04	0.4320
$f7$	**0**	0.2123	**0**	**0.1139**	**0**	0.5681	6.57E + 01	0.2711	1.84E + 02	0.3750
$f8$	**2.04E − 15**	0.1978	2.40E − 15	**0.1098**	3.41E − 15	0.5659	1.60E − 04	0.2610	3.23E + 00	0.3799
$f9$	**0**	0.2016	**0**	**0.1076**	**0**	0.6759	2.30E − 03	0.2927	2.11E − 02	0.4254
$f10$	4.39E − 10	0.1685	3.93E − 10	**0.0945**	**1.35E − 10**	0.8494	6.37E − 02	0.3409	5.52E + 00	0.6659

Table 3: Comparison between the proposed algorithms and the state-of-the-art algorithms (128D).

Func.	PWOA-PPM		PWOA-AFPM		WOA		CSO		PSO	
	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)
$f1$	**8.37E − 111**	**0.2614**	2.52E − 110	0.2781	1.21E − 110	1.1735	4.79E − 07	0.6871	1.08E − 01	1.0877
$f2$	**3.26E − 64**	**0.2588**	4.51E − 63	0.2761	1.71E − 62	1.1620	4.80E − 04	0.6495	4.05E + 02	1.1288
$f3$	2.40E + 03	**0.2584**	**1.68E + 03**	0.2817	2.45E − 03	1.1284	4.30E + 04	0.6805	9.83E + 04	1.3444
$f4$	**1.46E − 18**	**0.2540**	2.67E − 17	0.2693	1.70E − 17	1.1045	3.79E + 01	0.6723	5.37E + 01	1.1271
$f5$	4.36E − 01	0.2760	9.40E − 01	**0.2584**	**7.56E − 02**	1.1548	4.51E + 02	0.7085	7.63E + 02	1.1259
$f6$	−5.33E + 04	0.3525	**−5.35E + 04**	**0.3318**	−6.37E + 03	2.6482	−4.02E + 04	1.1221	V3.23E + 04	1.9468
$f7$	**0**	0.3321	**0**	**0.2641**	**0**	2.21	1.26E + 02	1.0309	4.43E + 02	1.7101
$f8$	4.58E − 15	0.3020	**2.22E-15**	**0.2798**	2.81E − 15	2.2070	8.17E − 04	1.0271	7.93E + 00	1.6865
$f9$	**0**	0.3193	**0**	**0.2733**	**0**	2.6825	1.11E − 02	1.0924	2.85E − 01	1.8733
$f10$	3.37E − 10	**0.2695**	2.69E − 11	0.3795	**2.01E − 11**	3.3434	4.35E − 01	1.3632	2.31E + 01	2.5813

extent. Generally speaking, benchmarking results of WOA, PWOAs, CSO, and PSO prove the effectiveness of two parallel WOAs proposed in this work.

Concerning running time, two perspectives of function type and scale settings are considered. From the perspective of function type, since multimodal functions ($f6 \sim f10$) generally have higher arithmetic complexity than unimodal functions ($f1 \sim f5$) [27, 40], there is an obvious time gap existed between unimodal and multimodal functions optimized by all algorithms in tables. For classic algorithms, WOA and PSO have relatively close running time, especially for unimodal functions. This is because the two algorithms essentially have similar structure and complexity. CSO, on the contrary, maintains faster performance than WOA and

Table 4: Comparison between the proposed algorithms and the state-of-the-art algorithms (256D).

Func.	PWOA-PPM		PWOA-AFPM		WOA		CSO		PSO	
	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)
$f1$	**3.47E − 118**	**0.6760**	5.40E − 118	1.7067	4.72E − 118	4.4267	2.01E − 06	2.6118	7.82E + 02	4.3250
$f2$	**3.54E − 65**	**0.7065**	7.52E − 65	1.6059	5.59E − 65	4.5381	8.65E − 04	2.7718	8.64E + 02	4.5422
$f3$	**8.89E + 02**	**0.6839**	3.15E + 03	1.6452	2.87E + 03	4.4668	1.62E + 05	2.8792	4.13E + 05	4.5937
$f4$	7.75E − 21	**0.6947**	**2.20E − 21**	1.7199	1.04E − 20	4.3673	3.46E + 01	2.6202	6.84E + 01	4.5842
$f5$	3.81E + 00	**0.7995**	2.40E − 01	1.6395	**8.51E − 02**	4.5709	7.68E + 02	2.6535	2.0E + 05	4.9438
$f6$	−1.05E + 05	**1.0876**	**−1.07E + 05**	1.5272	−9.01E + 03	10.5611	−7.45E + 04	4.6981	−5.63E + 04	7.1954
$f7$	**0**	**1.1013**	**0**	1.6448	**0**	8.7863	1.93E + 02	4.1165	9.30E + 02	6.0118
$f8$	**1.06E − 15**	**0.9937**	2.98E − 15	1.6502	3.06E − 15	8.7506	8.34E − 04	4.0297	1.36E + 01	6.0041
$f9$	**0**	**1.0277**	**0**	1.6496	**0**	10.6604	1.31E − 03	4.3609	1.14E + 01	8.1001
$f10$	2.34E − 11	**0.8478**	**3.04E − 12**	1.6143	6.36E − 12	13.3073	5.20E − 01	5.2544	8.13E + 01	9.5807

Table 5: Comparison between the proposed algorithms and the state-of-the-art algorithms (512D).

Func.	PWOA-PPM		PWOA-AFPM		WOA		CSO		PSO	
	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)	Mean	Time (s)
$f1$	**6.50E − 123**	**2.5753**	9.92E − 123	4.5841	1.38E − 122	17.6789	1.14E − 04	10.2368	8.32E + 04	18.1490
$f2$	**1.91E − 68**	**2.5890**	4.37E − 68	4.8587	9.88E − 68	18.0246	1.34E − 01	10.8658	1.74E + 03	19.0045
$f3$	1.73E + 04	**2.4585**	5.70E + 03	4.6452	**2.65E + 03**	17.8125	1.38E + 08	10.4507	1.54E + 06	18.5338
$f4$	**8.08E − 23**	**2.5920**	1.08E − 22	4.8721	4.79E − 22	17.4709	3.86E + 01	10.0652	9.81E + 01	18.7573
$f5$	**1.31E + 00**	**2.8612**	5.06E + 00	3.8336	7.35E + 00	18.2380	1.11E + 03	10.8096	1.00E + 08	18.8939
$f6$	−2.14E + 05	**3.1862**	**−2.17E + 05**	5.9001	−1.43E + 04	42.3103	−1.17E + 05	19.2030	−9.31E + 04	30.1889
$f7$	**0**	**3.0358**	**0**	3.9192	**0**	35.2929	2.79E + 03	17.0799	2.81E + 03	29.1872
$f8$	**6.13E − 16**	**2.9920**	3.29E − 15	5.6349	2.70E − 15	34.9978	4.36E − 03	15.6751	1.72E + 01	28.9203
$f9$	**0**	**3.0299**	**0**	4.3163	**0**	42.7160	5.22E − 04	18.2628	7.59E + 02	29.9871
$f10$	3.53E − 12	**2.9357**	**1.09E − 12**	7.4716	5.49E − 12	51.1879	3.34E − 01	21.4533	9.64E + 06	34.8555

PSO, where the simple and low-complexity algorithmic structures of CSO play an important part. In terms of PWOA-PPM and PWOA-AFPM, different scales of the problem have little impact on the execution time of these two algorithms. Regarding the problem scale, the running time of all three algorithms is affected by benchmark dimension and population size without exception. Canonical WOA is sensitive to problem scale, and different scale settings will cause a large gap in running time. For PWOA-PPM, the difference in execution time for functions with 64 D and 128 D is minimal but shows a trend of rapid growth for the scale of 256 D and 512 D. On the contrary, the performance of PWOA-PPM is relatively stable. As the scale increases, it demonstrates a slow growth of running time for PWOA-PPM.

In brief, the proposed PWOA-PPM and PWOA-AFPM are executed more stable than WOA, which benefits from the hardware-accelerated performance of FPGA due to built-in dedicated arithmetic units and modular design of the pipeline.

5.3. Speedup Analysis. In this section, speedup is calculated based on the running time of different problem scales, given as follows:

$$\text{Speedup} = \frac{T_{\text{WOA}}}{T_{\text{PWOA}}}, \tag{10}$$

where T_{WOA} and T_{PWOA} denote the running time of serial WOA and FPGA implementation of parallel WOA, respectively.

Speedup produced by PWOAs for settling various benchmark functions is shown in Figure 7 and analyzed as follows. Note that PWOAs have a certain degree of execution improvement and the speedup in both PWOA-PPM and PWOA-AFPM in multimodal functions is better than that in unimodal functions with all problem scales. From Figure 7(a), for both unimodal and multimodal functions, the greater the dimension of search space becomes, the higher the speedup ratio WOA-PPM obtains. Moreover, WOA-PPM can hold noticeable acceleration when solving the most complex $f10$, and the maximum speedup reaches around 18x with dimension = 512 D. As for WOA-AFPM, it has been found in Figure 7(b) that WOA-AFPM exhibits unstable computational performance, where the speedup for all functions decreases in case of 256 D while it manifests relatively better acceleration in the cases of 128 D and 512 D. In addition, the speedup ratio obtained by WOA-AFPM for optimizing $f10$, contrary to WOA-PPM, shows a slight downward trend, as the problem scale increases. The maximum speedup produced by PWOA-AFPM can be up to 10x (solving $f9$ with dimension = 512 D).

Four bar graphs, depicted in Figure 8, are used here to intuitively make comparisons for the speedup between PWOA-PPM and PWOA-AFPM with different problem scales. In cases of small scale including 64 D and 128 D, the speedup of PWOA-PPM is not as good as PWOA-AFPM, especially when solving all functions in case of 64 D and $f5 \sim f9$ in case of 128 D. Note, however, that the running efficiency of PWOA-PPM steadily rises as the scale

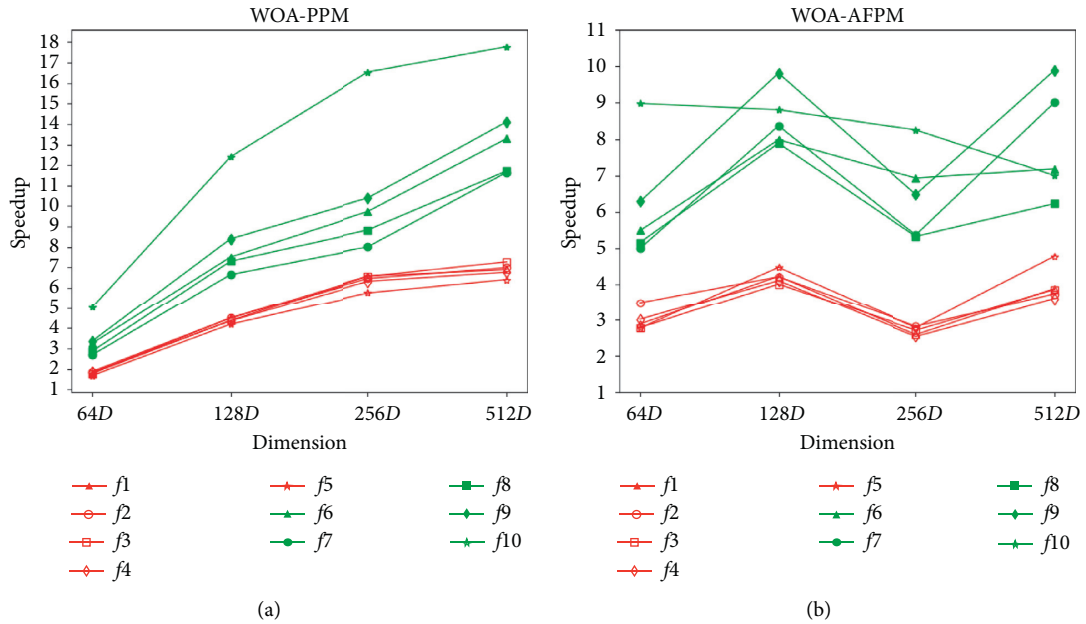


Figure 7: Speedup of PWOA-PPM and PWOA-AFPM w.r.t benchmark functions.

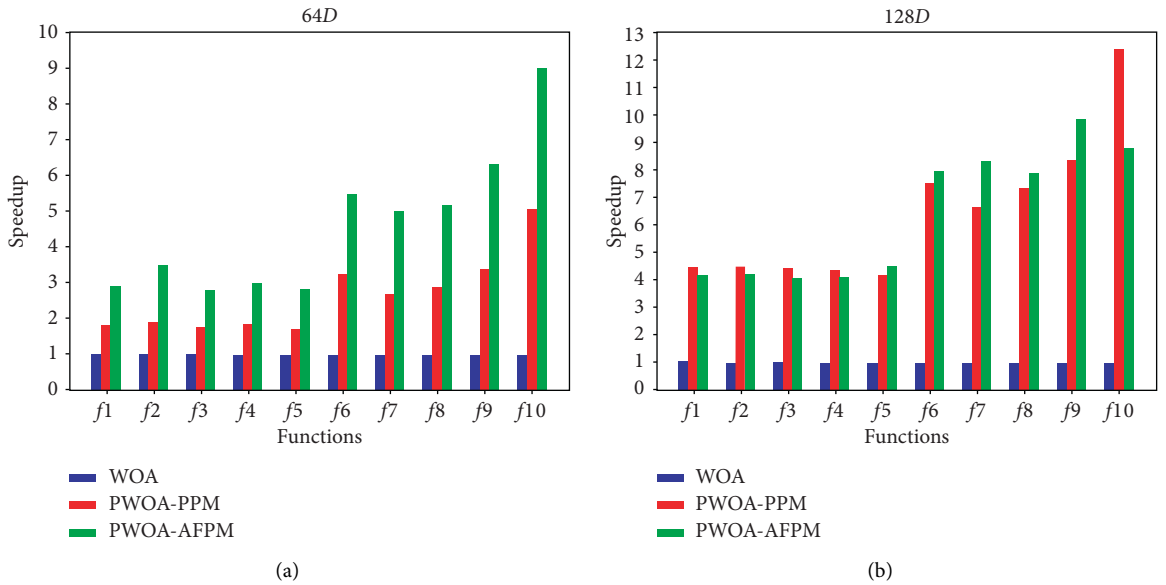


Figure 8: Continued.

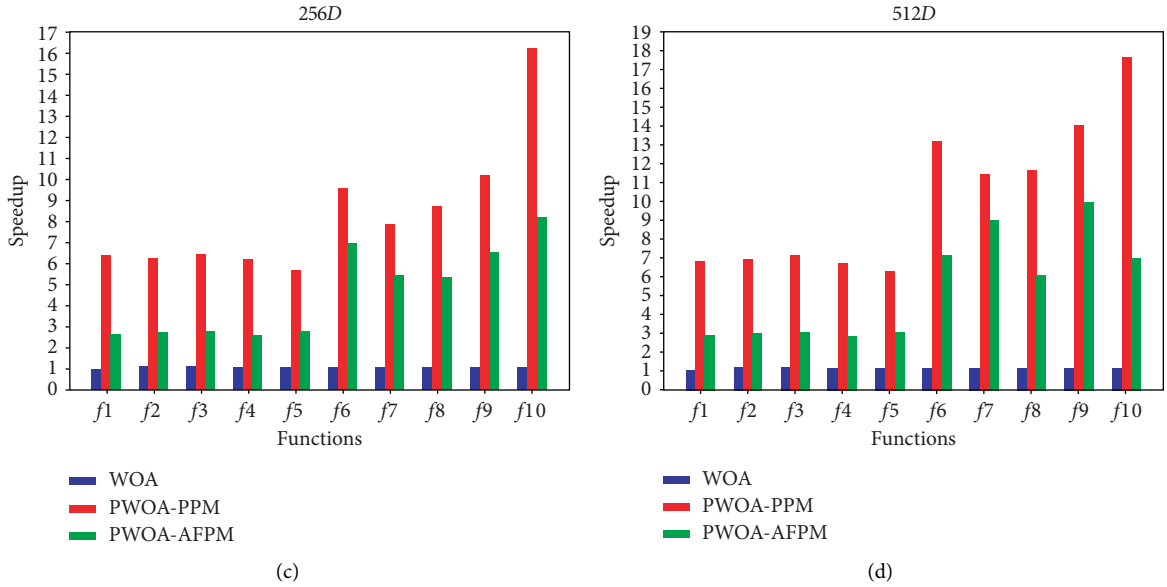


Figure 8: Comparison for the speedup between PWOA-PPM and PWOA-AFPM.

increases. In case of 256 D, the speedup of WOA-PPM for all unimodal functions and $f10$ is twice greater or more than that of WOA-AFPM. This value of speedup gap between WOA-PPM and WOA-AFPM further increases from twice to 2.5 times when solving $f10$ in case of 512 D. In a few words, WOA-PPM has more advantages in solving medium-scale and large-scale problems, while WOA-AFPM has better computational performance in small-scale problems.

It can be seen from above experimental analysis that PWOAs with two models have held discrepant influence on acceleration, which are mainly caused by the different frameworks instructing the implementation of PWOA-PPM and PWOA-AFPM on FPGA heterogeneous platform. For PWOA-PPM, it utilizes a partial parallel model and a certain amount of extra overhead that frequent communication between CPU and FPGA becomes a bottleneck leading to worse performance in case of small-scale. Unlike PWOA-PPM, PWOA-AFPM transfers most work of WOA to FPGA side for execution. Additionally, synchronous operation using memory fence requires more hardware to implement and might degrades kernel performance at FPGA side [55]. This, in turn, makes PWOA-AFPM become more inefficient with the increment of benchmark complexity and problem scale.

6. Conclusion

Demonstrating its excellence in global optimization, WOA has drawn significant research interests in the last few years. An unavoidable reality is that performance degradation takes places in WOA when facing large-scale complex optimization problems. Many proposals exist to address this issue, most of which, however, are based on classic algorithms such as genetic algorithm and particle swarm optimization, while very few literature studies about parallel WOA can be found. Based on FPGA accelerator, this study proposes two well-designed parallel models to implement parallel PWOA using the OpenCL framework with a demonstration on Intel heterogeneous platform. Finally, the performances of two parallel models based on PWOA (PWOA-PPM and PWOA-AFPM) have been evaluated using 10 benchmark functions.

For future work, it is essential to apply this algorithm to real engineering problems to verify the practical benefits. Besides, more different types of devices such as GPU and DSP need to be investigated, to build a multidevice heterogeneous platform. This platform will be an efficient cooperative running environment where a high-computational task can be decomposed into several parts and then assigned to different devices. Therefore, the proposed parallel scheme has potential for real applications.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work was supported by the National Key Research and Development Project under grant no. 2018YFB1700500, National Science Foundation of China under grant nos. 52077213, 62003332, and 61702493, Natural Science Foundation of Guangdong (no. 2018A030310671), and Outstanding Young Researcher Innovation Fund of Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences (no. 201822).

References

[1] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[2] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[3] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, p. 1, 2020.

[4] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 233–246, 2020.

[5] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang, "A hierarchical gravitational search algorithm with an effective gravitational constant," *Swarm and Evolutionary Computation*, vol. 46, pp. 118–139, 2019.

[6] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[7] R. Cheng, M. Li, K. Li, and X. Yao, "Evolutionary multiobjective optimization-based multimodal optimization: fitness landscape approximation and peak detection," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 692–706, 2018.

[8] H. Cheng, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Transactions on Cybernetics*, 2020.

[9] Y. Wang, Y. Yu, S. Cao, X. Zhang, and S. Gao, "A review of applications of artificial intelligent algorithms in wind farms," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3447–3500, 2020.

[10] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[11] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Generation Computer Systems*, vol. 85, pp. 129–145, 2018.

[12] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[13] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2018.

[14] J. Wang, P. Du, T. Niu, and W. Yang, "A novel hybrid system based on a new proposed algorithm-multi-objective whale optimization algorithm for wind speed forecasting," *Applied Energy*, vol. 208, pp. 344–360, 2017.

[15] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Multi-objective whale optimization algorithm for content-based image retrieval," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 26135–26172, 2018.

[16] A. Got, A. Moussaoui, and D. Zouache, "A guided population archive whale optimization algorithm for solving multi-objective optimization problems," *Expert Systems with Applications*, vol. 141, p. 112972, 2020.

[17] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Systems with Applications*, vol. 83, pp. 242–256, 2017.

[18] A. E. Hassanien, M. Abd Elfattah, S. Aboulenin, G. Schaefer, S. Y. Zhu, and I. Korovin, "Historic handwritten manuscript binarisation using whale optimisation," in *Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 003842–003846, IEEE, Budapest, Hungary, October 2016.

[19] A. Mostafa, A. E. Hassanien, M. Houseni, and H. Hefny, "Liver segmentation in MRI images based on whale optimization algorithm," *Multimedia Tools and Applications*, vol. 76, no. 23, pp. 24931–24954, 2017.

[20] S. Harikarthik, V. Palanisamy, and P. Ramanathan, "Optimal test suite selection in regression testing with testcase prioritization using modified ann and whale optimization algorithm," *Cluster Computing*, vol. 22, no. 5, pp. 11425–11434, 2019.

[21] S. Raj and B. Bhattacharyya, "Optimal placement of TCSC and SVC for reactive power planning using whale optimization algorithm," *Swarm and Evolutionary Computation*, vol. 40, pp. 131–143, 2018.

[22] H. M. Hasanien, "Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm," *Electric Power Systems Research*, vol. 157, pp. 168–176, 2018.

[23] S. Rahnamayan and G. G. Wang, "Solving large scale optimization problems by opposition-based differential evolution (ode)," *WSEAS Transactions on Computers*, vol. 7, no. 10, pp. 1792–1804, 2008.

[24] A. Jain, M. Mishra, S. K. Peddoju, and N. Jain, "Energy efficient computing-green cloud computing," in *Proceedings of 2013 International Conference on Energy Efficient Technologies for Sustainability*, pp. 978–982, IEEE, Nagercoil, April 2013.

[25] P. Liu, S. Li, and Q. Ding, "An energy-efficient accelerator based on hybrid CPU-FPGA devices for password recovery," *IEEE Transactions on Computers*, vol. 68, no. 2, pp. 170–181, 2018.

[26] Y. Zhou and Y. Tan, "Gpu-based parallel particle swarm optimization," in *Proceedings of 2009 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1493–1500, IEEE, Trondheim, Norway, May 2009.

[27] M. Jin and H. Lu, "Parallel particle swarm optimization with genetic communication strategy and its implementation on GPU," in *Proceedings of 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, pp. 99–104, IEEE, Hangzhou, China, October 2012.

[28] D. Narjess and B. Sadok, "A new hybrid GPU-PSO approach for solving max-csps," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 119-120, Denver, CO, USA, July 2016.

[29] M. P. Wachowiak, M. C. Timson, and D. J. DuVal, "Adaptive particle swarm optimization with heterogeneous multicore parallelism and GPU acceleration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2784–2793, 2017.

[30] J. Kumar, L. Singh, and S. Paul, "GPU based parallel cooperative particle swarm optimization using C-CUDA: a case study," in *Proceedings of 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, IEEE, Hyderabad, July 2013.

[31] J. Hajewski and S. Oliveira, "Two simple tricks for fast cache-aware parallel particle swarm optimization," in *Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1374–1381, IEEE, Wellington, New Zealand, March 2019.

[32] B. A. Menezes, H. Kuchen, H. A. A. Neto, and F. B. de Lima Neto, "Parallelization strategies for GPU-based ant colony optimization solving the traveling salesman problem," in *Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3094–3101, IEEE, Wellington, New Zealand, March 2019.

[33] Y. Djenouri, D. Djenouri, A. Belhadi, P. Fournier-Viger, J. Chun-Wei Lin, and A. Bendjoudi, "Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases," *Information Sciences*, vol. 496, pp. 326–342, 2019.

[34] C. Jin and A. K. Qin, "A GPU-based implementation of brain storm optimization," in *Proceedings of 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2698–2705, IEEE, San Sebastian, Spain, June 2017.

[35] L. Ma, T. Zhang, R. Wang, G. Yang, and Y. Zhang, "Pbar: parallelized brain storm optimization for association rule mining," in *Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1148–1156, IEEE, Wellington, New Zealand, March 2019.

[36] B. Chen, B. Chen, H. Liu, and X. Zhang, "A fast parallel genetic algorithm for graph coloring problem based on CUDA," in *Proceedings of 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 145–148, IEEE, Xi'an, China, September 2015.

[37] Y. Ma and L. S. Indrusiak, "Hardware-accelerated parallel genetic algorithm for fitness functions with variable execution times," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pp. 829–836, Denver, CO, USA, July 2016.

[38] H. Rico-Garcia, J.-L. Sanchez-Romero, A. Jimeno-Morenilla, H. Migallon-Gomis, H. Mora-Mora, and R. V. Rao, "Comparison of high performance parallel implementations of TLBO and JAYA optimization methods on manycore GPU," *IEEE Access*, vol. 7, pp. 133822–133831, 2019.

[39] Y. Khalil, M. Alshayeji, and I. Ahmad, "Distributed whale optimization algorithm based on mapreduce," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 1, p. e4872, 2019.

[40] Y. Tan and K. Ding, "A survey on GPU-based implementation of swarm intelligence algorithms," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2028–2041, 2016.

[41] F. A. Escobar, X. Chang, and C. Valderrama, "Suitability analysis of FPGAs for heterogeneous platforms in HPC," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 600–612, 2015.

[42] H. M. Waidyasooriya, Y. Takei, S. Tatsumi, and M. Hariyama, "OpenCL-based FPGA-platform for stencil computation and its optimization methodology," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1390–1402, 2016.

[43] D. Weller, F. Oboril, D. Lukarski, J. Becker, and M. Tahoori, "Energy efficient scientific computing on FPGAs using OpenCL," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 247–256, Monterey, CF, USA, February 2017.

[44] K. Shata, M. K. Elteir, and A. A. EL-Zoghabi, "Optimized implementation of OpenCL kernels on FPGAs," *Journal of Systems Architecture*, vol. 97, pp. 491–505, 2019.

[45] D. Li, L. Huang, K. Wang, W. Pang, Y. Zhou, and R. Zhang, "A general framework for accelerating swarm intelligence algorithms on FPGAs, GPUS and multi-core CPUS," *IEEE Access*, vol. 6, pp. 72327–72344, 2018.

[46] H. M. Waidyasooriya, M. Hariyama, M. J. Miyama, and M. Ohzeki, "OpenCL-based design of an FPGA accelerator for quantum annealing simulation," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 5019–5039, 2019.

[47] Q. Jiang, Y. Guo, Z. Yang, and X. Zhou, "A parallel whale optimization algorithm and its implementation on FPGA," in *Prceedings of 2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, Glassgow, UK, July 2020.

[48] *OpenCL Overview*, 2019 https://www.khronos.org/opencl/</u>.

[49] *Intel FPGA SDK for OpenCL*, https://www.intel.com/content/www/us/en/software/programmable/sdk-for-opencl/overview.html</u>, 2019.

[50] D. Chen and D. Singh, "Using OpenCL to evaluate the efficiency of CPUS, GPUS and FPGAs for information filtering," in *Proceedings of 22nd International Conference on Field Programmable Logic and Applications (FPL)*, pp. 5–12, IEEE, Oslo, Norway, August 2012.

[51] J. Zhang and J. Li, "Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 25–34, Monterey, MX, USA, February 2017.

[52] U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An openclTM deep learning accelerator on arria 10," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 55–64, Monterey, MX, USA, February 2017.

[53] N. Suda, V. Chandra, G. Dasika et al., "Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 16–25, Monterey, CF, USA, February 2016.

[54] *Intel Fpga Sdk for Opencl Pro Edition: Programming Guide*, 2019 https://www.intel.cn/content/dam/www/programmable/us/en/pdfs/literature/hb/opencl-sdk/aocl_programming_guide.pdf</u>.

[55] *Intel Fpga Sdk for Opencl Pro Edition: Best Practices Guide*, 2019 https://www.intel.cn/content/dam/www/programmable/us/en/pdfs/literature/hb/opencl-sdk/aocl-best-practices-guide.pdf</u>.

[56] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.

[57] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, UK, November 1995.

[58] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2014.

Hindawi
Complexity
Volume 2020, Article ID 6691764, 15 pages
https://doi.org/10.1155/2020/6691764

Research Article

Operational Safety Risk Assessment for the Water Channels of the South-to-North Water Diversion Project Based on TODIM-FMEA

Huimin Li[1], **Li Ji**[1], **Feng Li**[1], **Hairui Li**[2], **Qingguo Sun**[3], **Zhihong Li**[3], **Hongmei Yan**[3], **Wei Guan**[3], **Lunyan Wang**[1,4,5] and **Ying Ma**[1,4]

[1]*Department of Construction Engineering and Management, North China University of Water Resources and Electric Power, Zhengzhou 450046, China*
[2]*School of Management and Economics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China*
[3]*Bureau of South to North Water Transfer of Planning, Design and Management, Ministry of Water Resources, Beijing 100038, China*
[4]*Henan Key Laboratory of Water Environment Simulation and Treatment, Zhengzhou 450045, China*
[5]*Collaborative Innovation Center of Water Resources Efficient Utilization and Protection Engineering, Zhengzhou 450045, China*

Correspondence should be addressed to Feng Li; lifeng9406@126.com

Received 26 October 2020; Revised 26 November 2020; Accepted 30 November 2020; Published 12 December 2020

Academic Editor: Shi Cheng

The South-to-North Water Diversion Project consists of long-distance water delivery channels and a complicated geological environment along the way. To deal with the operation safety of the water conveyance channels in the middle route of the South-to-North Water Diversion Project, this study analyzes six failure modes: structural cracks, poor water delivery during ice periods, instability of canal slopes, material aging, abnormal leakage, and foundation defects. Based on FMEA, a multigranularity language evaluation method that can be converted into interval intuitionistic fuzzy numbers is used to evaluate the severity (S), occurrence (O), and detection difficulty (D) of the six failure modes. Interval intuitionistic fuzzy entropy is used to calculate the weights of the risk factors. Finally, a ranking model of each failure mode is built based on the TODIM method. The final ranking results show that the risk of abnormal leakage is the largest, and the risk of poor water delivery during ice periods is the smallest. The feasibility and validity of the calculation results are verified by comparing them with the ranking results of the traditional RPN and TOPSIS methods. The TODIM-FMEA risk assessment model offers a new solution to the problem of risk assessment for water transfer projects.

1. Introduction

The South-to-North Water Diversion Project (SNWDP), also called the South-North Water Transfer Project, is an ongoing Chinese effort to channel 45 billion m^3 of water annually from the Yangtze River in southern China to the country's less-fertile northern regions [1]. The SNWDP consists of three water diversion projects, the East Route Project (ERP), the Middle Route Project (MRP), and the West Route Project (WRP), which divert water from the lower, middle, and upper reaches of the Yangtze River, respectively, as shown in Figure 1. Both Phase I of the ERP and Phase I of the MRP have been in operation since 2013 and 2014, respectively [2]. The 1432 km MRP diverts water from the Danjiangkou Reservoir in the Hanjiang River basin to 20 major cities and 100 counties in Henan and Hebei provinces, Beijing, and Tianjin municipalities [3]. There are complex hydraulic structures in the long channel, and MRP provides drinking water in an open/covered channel to Beijing and Tianjin under extremely strict water quality requirements, all of which pose a significant challenge [4, 5]. The average annual water supply capacity of the MRP is 9.5 billion cubic meters [6]. The MRP mainly comprises open channels for water

Figure 1: The three routes of the South-to-North Water Diversion Project.

conveyance, and the complex geological conditions and natural environment along the route make the potential operation risks very complex and difficult to grasp. In order to effectively control the operation risks, ensure the water diversion system can operate safely, efficiently, and scientifically, and maximize the benefits of the project, it is necessary to carry out risk identification and risk assessment in advance.

Many scholars have conducted extensive research on the operation risk of the South-to-North Water Diversion Project. Du and Geng [7] adopted risk projection graphs to predict and analyze the engineering risks of the water conveyance channels in the MRP and used failure probabilities and grades of failure consequences to evaluate the risks of each engineering section of the water conveyance channels in the MRP. Hu et al. [8] summarized the operation risk evaluation indices of open water channels and obtained risk grades by using right-angle fuzzy sets and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [9]. Zhou et al. [10] put forward a plan and management measures for dispatching water during ice periods and formulated an emergency plan and early warning forecast for ice periods. Cheng et al. [9] believed that the water diversion project is a long line series system and calculated the system reliability indices of the three typical modes of failure, i.e., overtopping, seepage failure, and landslide of channel slope failures in the MRP, by approximately reconstructing a linear system safety margin equation. Xiong et al. [11] classified and summarized the failure modes of different hydraulic structures in the MRP. To sum up, the risk analysis studies of typical hydraulic structures in the MRP have been widely carried out, but most of them are qualitative analyses or quantitative evaluations of specific failure modes of typical hydraulic structures.

Failure Mode and Effects Analysis (FMEA) is a widely used method for analyzing system reliability and risk management in various fields, such as automotive, manufacturing, chemical, and engineering [12–14]. Traditional FMEA multiplies the three risk factors, i.e., severity (S), occurrence (O), and detection difficulty (D), to obtain the risk priority number (RPN) and then ranks the identified failure modes [12]. Although the traditional FMEA method has the advantages of ease of use and versatility, there are still many defects. Many scholars have used different methods to improve the FMEA method to make the results more reasonable. Chang [15] introduced binary semantic variables to assess the failure mode information. Vahdani et al. [16] proposed a FMEA method based on the fuzzy belief structure representation. Wang et al. [17] improved the FMEA method by using language variables convertible to intuitionistic fuzzy numbers to represent information regarding failure modes and the weight of the risk factors. These scholars resolved the shortcoming of the traditional FMEA method, which was the difficulty of representing failure mode information directly using real numbers.

Other scholars have resolved another defect of the traditional FMEA method, which is its failure to account for the weight information of the risk factors. Emovon et al. [18] proposed an objective weighting method based on a synthesis of the variance method and entropy weight method. Zhu et al. [19] used the TOPSIS method to build an optimization model of risk factor weights. Many scholars introduced multicriteria decision-making methods to improve FMEA. Liu et al. [20] proposed a method based on intuitionistic fuzzy numbers and TOPSIS to improve the sorting of failure modes in FMEA. Chang et al. [21] considered the correlation between failure modes and proposed a FMEA

risk assessment method based on Decision-Making Trial and Evaluation Laboratory (DEMATEL) and TOPSIS. These methods resolved the problem with traditional RPN value results, that is, identical risk priority values may be obtained, which make failure mode risk ranking unworkable. The interactive multicriteria decision-making method (Tomada de Decisão Interativa Multicritério, TODIM) is a multicriteria decision-making method based on the prospect theory proposed by Gomes and Lima [22]. In this method, the psychological behaviors of decision makers are taken into full consideration, and alternatives are ranked according to the relative dominance of one alternative over other alternatives.

However, it should be pointed out that there are still the following problems with the existing studies: (1) the existing information assessment methods adopted by representation experts are useful in characterizing the fuzziness and uncertainty of the information, but they fail to consider the certainty level, uncertainty level, and hesitation level simultaneously; (2) the existing multicriteria decision-making methods resolve the problem with the traditional methods, that is, identical RPN values, which make ranking unworkable, but they neglect the influence of the psychological behaviors of different experts on the risk ranking results.

In order to overcome the shortcomings of FMEA, in this paper, multigranularity language information is converted into interval intuitionistic fuzzy numbers to represent the failure modes. The weights of risk factors are calculated using improved interval intuitionistic fuzzy entropy, which has been adapted to a TODIM-based FMEA method to rank the failure modes. The improved FMEA method is applied to assess operation safety risks of the MRP.

The remainder of the paper is organized as follows. Section 2 presents operation risks and failure modes of the water channels in the MRP. Section 3 introduces the research methods: a TODIM-FMEA risk evaluation method based on multigranularity language information. A case study is presented in Section 4. Finally, conclusions and recommendations are drawn in Section 5.

2. Failure Modes of the Water Channels in the MRP

The majority of the open water channels in the MRP is made of lined concrete structures, and lined concrete structures have the characteristics of being thin and having large surface areas, as shown in Figure 2. The distance along the channels is quite long, and the channels pass through high embankments, deep excavations, and expansive soil treatment sections. The water channels are highly exposed to the natural environment and human activities, which make the operation safety risks more complex than they are in any other channels. Based on an analysis of the risk factors in the literature [7–11] and on the opinions of practical experts on the MRP, this paper has chosen the six most typical failure modes of the water conveyance channels. The failure modes include structural cracks (FM1), poor water delivery during ice periods (FM2), instability of canal slopes (FM3), material aging (FM4), abnormal leakage (FM5), and foundation defects (FM6). The main causes and failure consequences of the six failure modes are shown in Table 1.

Figure 2: Water channel of the middle route of the South-to-North Water Diversion Project.

2.1. Structural Cracks (FM1). Cracks are common in concrete structures, and they are caused by the interaction between the concrete and the external environment. The fundamental reason for cracks is when structural stress exceeds the tensile strength of the concrete materials. Cracks often appear in various structures. When a crack is not serious, it will not affect the normal operation of the project, but cracks accelerate the aging of the concrete and reduce the structure's ability to resist freeze-thaw damage and environmental erosion damage. In channel engineering, cracks have very limited effect on structures when they appear initially, but if no measures are taken to prevent the cracks from expanding, they will cause leakage and serious damage to the lining of the structures.

2.2. Poor Water Delivery during Ice Periods (FM2). Almost half of the MRP is in the northern part of China, where the winter temperatures are always below zero degrees centigrade, and the channels thus deliver water with ice. The shape of the cross sections of the channels through which water flows is an inverted trapezoid, and the cross sections may be affected by the flowing ice and ice jams. The phenomenon of backwater may form due to ice jams, increasing the danger of water overtopping the embankments. During ice periods, it is also possible that the ice slag mixed with floating grass may endanger the safe operation of the water delivery system. During ice periods, the variation of the water level should be strictly controlled, and the water flow should be reduced properly to keep the flow rate at a low level.

2.3. Instability of the Channel Slope (FM3). The causes of channel instability risks mainly include foundation conditions and environment conditions. Foundation conditions primarily refer to the geological conditions of the terrain through which the channels pass. If there are unstable properties and weak intercalations in the foundation, the

Table 1: The failure modes of the water channels in the MRP during the operation stage.

Serial number	Failure mode	Main causes of the failure	Effects of the failure
FM1	Structural cracks	Temperature change, shrinkage, freezing, uneven settling of the foundation, etc.	Reduce the durability of the structure, weaken its bearing capacity, and can cause failure and destruction of the structures
FM2	Poor water delivery during ice periods	Ice blocks the water delivery	Water overflows and damage to channel embankments
FM3	Instability of the channel slope	Sudden change of water level; reduction of the antislide stability of the foundation	Causes overtopping and regional floods around the water diversion system
FM4	Material aging	Carbonization of concrete, temperature change, and freeze-thaw damage	Leakage
FM5	Abnormal leakage	Microcracks and biological destruction	The collapse of the bank slope, which affects the safety of the structures
FM6	Foundation defects	Expansive (rock) soils, loess soil, saturated sand sections, and filled sections	Loss of bearing capacity of the foundation and collapse of structures

Note. FM: failure mode.

channels will be prone to instability and deformation. Environment conditions primarily refer to earthquakes, torrential rain, flooding, snowmelt, and sudden changes in the water level. These environmental factors affect the stress state of the embankment and cause instability of the channel slope.

2.4. Material Aging (FM4). Material aging is a common phenomenon in water conservancy projects. The chief causes of material aging are concrete carbonization, temperature changes, and freeze-thaw damage. Firstly, carbonization occurs when the alkaline material in concrete reacts with CO_2, producing carbonate and water, and this process will reduce the strength of the concrete. Carbonization causes the protective layers to deteriorate and steel bars to corrode, which further weakens the strength of the concrete. Secondly, temperature changes cause the shrinkage of concrete and uneven subsidence, which lead to cracks in the concrete. Finally, freeze-thaw damage is typically manifested in the shedding and cracking of the concrete lining of the channels, which leads to serious water leakage.

2.5. Abnormal Leakage (FM5). The embankment of the MRP is made of soil, so leakage from the channels erodes the foundation, endangering the overall stability of the channel structure. In water conservancy projects, the dryness and wetness of the environment take effect alternately, which leads to the concrete alternating between shrinkage and expansion and produces microcracks. The expansion and penetration of microcracks lead to significant seepage from the concrete. In addition, uneven subsidence deformation and biological damage cause cracks. For instance, mice or termite nests will also directly cause water seepage, piping, and, at worst, subsidence.

2.6. Foundation Defects (FM6). The MRP consists of expansive rock, expansive soil, loessial soil, saturated sand, and filled sections. One characteristic of expansive soil is that it expands when encountering water and shrinks when losing water. The presence of expansive soil causes the channels to crack, tilt, and even be destroyed. Most loessial soil is collapsible when it encounters water, which results in the subsidence of the channel foundation and slopes. Saturated sand always creates the problem of liquefaction. Liquefaction reduces the bearing capacity of the foundation, eventually causing the structures to subside and collapse. In the filled section of the channel, if the filling is not compacted, it results in instability and leakage.

3. Research Method

In this study, a multigranularity linguistic evaluation method that can be converted into interval intuitionistic fuzzy numbers is used to evaluate the severity (S), occurrence (O), and detection difficulty (D) in FMEA. Interval intuitionistic fuzzy entropy is used to calculate risk factor weights. And, finally, a ranking model of each failure mode is built based on the TODIM method.

3.1. Risk Expression Based on Multigranularity Language Information Convertible to Interval Intuitionistic Fuzzy Numbers

3.1.1. Multigranularity Language Information

Definition 1. The multigranularity language assessment set is denoted as

$$S^q = \left\{ s_i^q | i \in \left| \left\{ -\frac{q-1}{2}, \ldots, -1, 0, 1, \ldots, \frac{q-1}{2} \right\} \right. \right\}. \quad (1)$$

In the formula, S_i^q indicates the i evaluation phrase of S^q, and q must be odd. When $q = 5$, the evaluation phrase set $S^q$is denoted as {very bad, bad, average, good, very good}.

Generally, the properties of S_i^q should meet the following requirements:

(1) Order: when $i \geq j$, $s_i^q \geq s_j^q$

(2) Inverse operator: when $j = q - 1$, Neg $(s_i^q) = s_i^q$

(3) Maximum operation: if $s_i^q \geq s_i^q$, then $\max(s_i^q, s_j^q) = s_i^q$

(4) Minimal operation: if $s_i^q \le s_i^q$, then $\max(s_i^q, s_j^q) = s_i^q$

3.1.2. Intuitionistic Fuzzy Sets

Definition 2 (see [23]). X is a nonempty set, $X = \{x_1, x_2, \ldots, x_n\}$, and, on X, each of those like $A = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in X\}$ is called an intuitionistic fuzzy set on X, where $\mu_A(x)$ and $\nu_A(x)$ are the membership grade and nonmembership grade of element x belonging to judgment set A, where $\mu_A(x)$: $X \longrightarrow [0,1]$, $\nu_A(x)$: $X \longrightarrow [0,1]$, and $0 \le \mu_A(x) + \nu_A(x) \le 1$, $x \in X$.

In addition, $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ indicates the hesitancy degree of the element x belonging to A, where $0 \le \pi_A(x) \le 1$, $x \in X$.

Definition 3 (see [24]). When X is set as a nonempty set, $X = \{x_1, x_2, \ldots, x_n\}$, then an interval intuitionistic fuzzy set on X is defined as $\widetilde{A} = \{\langle x, \mu_{\widetilde{A}}(x), \nu_{\widetilde{A}}(x)\rangle | x \in X\}$, where $\mu_{\widetilde{A}}$: $X \longrightarrow [0,1]$ and $\nu_{\widetilde{A}}$: $X \longrightarrow [0,1]$, and for all $x \in X$, meets $0 \le \mu_{\widetilde{A}}(x) + \nu_{\widetilde{A}}(x) \le 1$. Interval number $\mu_{\widetilde{A}}(x)$ and interval number $\nu_{\widetilde{A}}(x)$ indicate the membership grade and nonmembership grade of element x belonging to A, respectively.

In addition, $\pi_{\widetilde{A}}(x) = 1 - \mu_{\widetilde{A}}(x) - \nu_{\widetilde{A}}(x)$, which is called the intuition index of the element x in A, which is the intuitionistic fuzzy set, indicates the hesitancy degree of the element x belonging to $\widetilde{A}$, where $0 \le \pi_{\widetilde{A}}(x) \le 1$, $x \in X$. For each $x \in X$, $\mu_{\widetilde{A}}(x)$ and $\nu_{\widetilde{A}}(x)$ are closed interval numbers, and their lower bound and the upper bound are denoted as $\mu_{\widetilde{A}}^L(x)$, $\mu_{\widetilde{A}}^U(x)$, $\nu_{\widetilde{A}}^L(x)$, and $\nu_{\widetilde{A}}^U(x)$, respectively.

For convenience, the intuitionistic fuzzy set A can be denoted as

$$\widetilde{A} = \left\{\langle x, \left[\mu_{\widetilde{A}}^L(x), \mu_{\widetilde{A}}^L(x)\right], \left[\nu_{\widetilde{A}}^L(x), \nu_{\widetilde{A}}^L(x)\right]\rangle | x \in X\right\}, \quad (2)$$

where $0 \le \mu_{\widetilde{A}}^U(x) + \nu_{\widetilde{A}}^U(x) \le 1$, $0 \le \mu_{\widetilde{A}}^L(x) \le \mu_{\widetilde{A}}^L(x) \le 1$, and $0 \le \nu_{\widetilde{A}}^L(x) \le \nu_{\widetilde{A}}^U(x) \le 1$. In addition, $\pi_{\widetilde{A}}(x) = 1 - \mu_{\widetilde{A}}(x) - \nu_{\widetilde{A}}(x) = [1 - \mu_{\widetilde{A}}^U(x) - \nu_{\widetilde{A}}^U(x), 1 - \mu_{\widetilde{A}}^L(x) - \nu_{\widetilde{A}}^L(x)]$.

Definition 4 (see [25]). Set $\widetilde{A} = (a_1, a_2], [a_3, a_4])$ as an interval intuitionistic fuzzy number; then, $S(\widetilde{A}) = (a_1 - a_2 + a_3 - a_4)/2$ is called the score function of the interval intuitionistic fuzziness.

Set $\widetilde{A} = (a_1, a_2], [a_3, a_4])$ and $\widetilde{B} = (b_1, b_2], [b_3, b_4])$ as two interval intuitionistic fuzzy numbers: if $S(\widetilde{A}) < S(\widetilde{B})$, then $\widetilde{A} < \widetilde{B}$.

3.1.3. Convert Multigranularity Language Phrases into Interval Intuitionistic Fuzzy Numbers

Definition 5. Set $S^q = \{s_i^q | i \in |\{-(q-1)/2, \ldots, -1, 0, 1, \ldots, (q-1)/2\}\}$ as a language evaluation set; s_i^q is inside S^q and can be converted into interval intuitionistic fuzzy numbers as

$$S^q = \left\{(\widetilde{\mu}_i^q, \widetilde{\nu}_i^q) | i \in |\left\{-\frac{q-1}{2}, \ldots, -1, 0, 1, \ldots, \frac{q-1}{2}\right\}\right\}, \quad (3)$$

where

$$\widetilde{\mu}_i^q = \begin{cases} (\widetilde{\mu}_0^q)^{1-i}, & i < 0, \\ (\widetilde{\mu}_0^q)^{\frac{1}{1+i}}, & i \ge 0 \end{cases}$$

$$\widetilde{\nu}_i^q = \begin{cases} 1 - (1 - \widetilde{\nu}_0^q)^{1-i}, & i < 0, \\ 1 - (1 - \widetilde{\nu}_0^q)^{\frac{1}{1+i}}, & i \ge 0, \end{cases} \quad (4)$$

$$\widetilde{\mu}_0^q = \widetilde{\nu}_0^q = \left[0.5 - \frac{1}{2q}, 0.5\right].$$

There are many ways to calculate the distance between interval intuitionistic fuzzy numbers. This paper adopts a new distance formula by combining the two traditional distance measurement formulas, i.e., Hamming distance and Hausdorff distance. The formula combines the advantages of the two distance measures, defined as follows.

Definition 6. Set $\widetilde{A} = \{\langle x, [\mu_{\widetilde{A}}^L(x), \mu_{\widetilde{A}}^U(x)], [\nu_{\widetilde{A}}^L(x), \nu_{\widetilde{A}}^U(x)]\rangle | x \in X\}$, $\widetilde{B} = \{\langle x, [\mu_{\widetilde{B}}^L(x), \mu_{\widetilde{B}}^U(x)], [\nu_{\widetilde{B}}^L(x), \nu_{\widetilde{B}}^U(x)]\rangle | x \in X\}$, and $\widetilde{C} = \{\langle x, [\mu_{\widetilde{C}}^L(x), \mu_{\widetilde{C}}^U(x)], [\nu_{\widetilde{C}}^L(x), \nu_{\widetilde{C}}^U(x)]\rangle | x \in X\}$ as three IVIFNs; then, the distance between $\widetilde{A}$ and $\widetilde{B}$ is defined as

$$\begin{aligned} d(\widetilde{A}, \widetilde{B}) = {} & \frac{1}{8}\sum_{i=1}^{n}\Big(\left|\mu_{\widetilde{A}}^L(x_i) - \mu_{\widetilde{B}}^L(x_i)\right| + \left|\mu_{\widetilde{A}}^U(x_i) - \mu_{\widetilde{B}}^U(x_i)\right| \\ & + \left|\nu_{\widetilde{A}}^L(x_i) - \nu_{\widetilde{B}}^L(x_i)\right| + \left|\nu_{\widetilde{A}}^U(x_i) - \nu_{\widetilde{B}}^U(x_i)\right| \\ & + \left|\pi_{\widetilde{A}}^L(x_i) - \pi_{\widetilde{B}}^L(x_i)\right| + \left|\pi_{\widetilde{A}}^U(x_i) - \pi_{\widetilde{B}}^U(x_i)\right|\Big) \\ & + \frac{1}{2}\sum_{i=1}^{n}\Big(\max\Big(\left|\mu_{\widetilde{A}}^L(x_i) - \mu_{\widetilde{B}}^L(x_i)\right|, \left|\mu_{\widetilde{A}}^U(x_i) - \mu_{\widetilde{B}}^U(x_i)\right|, \\ & \left|\nu_{\widetilde{A}}^L(x_i) - \nu_{\widetilde{B}}^L(x_i)\right|, \left|\nu_{\widetilde{A}}^U(x_i) - \nu_{\widetilde{B}}^U(x_i)\right|, \\ & \left|\pi_{\widetilde{A}}^L(x_i) - \pi_{\widetilde{B}}^L(x_i)\right|, \left|\pi_{\widetilde{A}}^U(x_i) - \pi_{\widetilde{B}}^U(x_i)\right|\Big), \end{aligned} \quad (5)$$

where $\pi_{\widetilde{A}}^L(x_i) = 1 - \mu_{\widetilde{A}}^U(x_i) - \nu_{\widetilde{A}}^U(x_i)$, $\pi_{\widetilde{A}}^U(x_i) = 1 - \mu_{\widetilde{A}}^L(x_i) - \nu_{\widetilde{A}}^L(x_i)$, $\pi_{\widetilde{B}}^L(x_i) = 1 - \mu_{\widetilde{B}}^U(x_i) - \nu_{\widetilde{B}}^U(x_i)$, and $\pi_{\widetilde{B}}^U(x_i) = 1 - \mu_{\widetilde{B}}^L(x_i) - \nu_{\widetilde{B}}^L(x_i)$.

3.2. TODIM-FMEA Risk Assessment Method Based on Multigranularity Language Information. The TODIM-FMEA risk assessment model based on multigranularity language information consists of three steps.

(1) FMEA risk assessment experts identify the failure modes and assess the risk factors of the water channels in the MRP. (2) The weights of risk factors S, O, and D are

calculated using interval intuitionistic fuzzy entropy. (3) The interval intuitionistic fuzzy TODIM method is used to rank the risk degrees of the failure modes. The framework of the proposed FMEA model is shown in Figure 3.

In the actual assessment process, information related to failure modes has great uncertainty and incompleteness; therefore, experts must refer to historical data about water channels actually in the MRP to evaluate failure modes. Experts use language variables to express the evaluation information. Then, the language variables are converted into interval intuitionistic fuzzy sets to carry out the calculations. The proposed FMEA model steps are as follows:

Step 1: the expert scoring team comprises s experts E_k $(k = 1, 2, \ldots, s)$, and different weights are assigned to them based on their level of expertise. The risk evaluation process will focus on m potential failure modes FM_i $(i = 1, 2, \ldots, m)$ of the three risk factors, i.e., severity (S), occurrence (O), and detection (D). The assessment of the three risk factors is presented by multigranularity language information, and then the language variable assessment matrix $X = [x_{ij}]_{m\times n}$ is determined. The evaluation results should be converted into the corresponding interval intuitionistic fuzzy numbers, where $R = [a_{ij}]_{m\times n}$ indicates the failure mode interval intuitionistic fuzzy evaluation matrix given by the s^{th} expert. Based on the weight assigned to the experts and the interval-valued intuitionistic fuzzy weighted arithmetic averaging (IIWAA) operator, the interval intuitionistic fuzzy evaluation matrix is converted into the group integrated weighted interval-valued intuitionistic fuzzy judgment matrix, which is $R' = [a_{ij}]_{m\times n}$.

Definition 7 (see [26]). Set $\widetilde{\alpha}_j = ([\widetilde{a}_j, \widetilde{b}_j]_j, [\widetilde{c}_j, \widetilde{d}_j])$ as the interval intuitionistic fuzzy numbers, $j = 1, 2, \ldots, n$, and IIWAA: $\Omega^n \longrightarrow \Omega$ if

$$\mathrm{IIWAA}_{\omega}(\widetilde{\alpha}_1, \widetilde{\alpha}_2, \ldots, \widetilde{\alpha}_n) = \sum_{j=1}^{n} \omega_j \alpha_j = \left\langle \left[1 - \prod_{j=1}^{n}\left(1 - a_j\right)\omega_j, 1 - \prod_{j=1}^{n}\left(1 - b_j\right)\omega_j\right], \left[\prod_{j=1}^{n} c_j\omega_j, \prod_{j=1}^{n} d_j\omega_j\right]\right\rangle, \tag{6}$$

where $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ are weight vectors and $\omega_j \in [0, 1]$, $\sum_{j=1}^{n} \omega_j = 1$; then, we call IIWAA the weighted arithmetic integration operator of the interval-valued intuitionistic numbers.

Step 2: use improved interval intuitionistic fuzzy entropy to calculate the weights and relative weights of risk factors S, O, and D.

Definition 8. For any interval intuitionistic fuzzy set $\widetilde{A} \in \mathrm{IVIFS}\, s(X)$, $\forall x \in X$, its entropy is

$$E(\widetilde{A}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\left[1 - \frac{1}{2}\left(\left|\mu_{\widetilde{A}}^{L}(x_i) - \nu_{\widetilde{A}}^{L}(x_i)\right| + \left|\mu_{\widetilde{A}}^{U}(x_i) - \nu_{\widetilde{A}}^{U}(x_i)\right|\right)\right]\ln\left[\frac{\pi_{\widetilde{A}}^{L}(x_i) + \pi_{\widetilde{A}}^{U}(x_i)}{2}\right]}{\left(1 - \frac{1}{2}\left(\left|\mu_{\widetilde{A}}^{L}(x_i) - \nu_{\widetilde{A}}^{L}(x_i)\right| + \left|\mu_{\widetilde{A}}^{U}(x_i) - \nu_{\widetilde{A}}^{U}(x_i)\right|\right) + e - 1\right]}, \tag{7}$$

where $(1 - (1/2)(|\mu_{\widetilde{A}}^{L}(x_i) - \nu_{\widetilde{A}}^{L}(x_i)| + |\mu_{\widetilde{A}}^{U}(x_i) - \nu_{\widetilde{A}}^{U}(x_i)|)$ is the level of information insufficiency; $(1/2)(|\mu_{\widetilde{A}}^{L}(x_i) - \nu_{\widetilde{A}}^{L}(x_i)| + |\mu_{\widetilde{A}}^{U}(x_i) - \nu_{\widetilde{A}}^{U}(x_i)|)$ is the interval distance of the degree of membership; $(\pi_{\widetilde{A}}^{L}(x_i) + \pi_{\widetilde{A}}^{U}(x_i))/2$ is the average value of the maximum degree of membership and the minimum degree of membership, as well as the level of information insufficiency; and e is the base of natural logarithms.

Based on improved interval intuitionistic fuzzy entropy formula (7), entropy matrix E is built on the basis of the group integrated weighted interval-valued intuitionistic fuzzy judgment matrix:

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix}. \tag{8}$$

After normalization, the standard entropy matrix is built as $\overline{E}$:

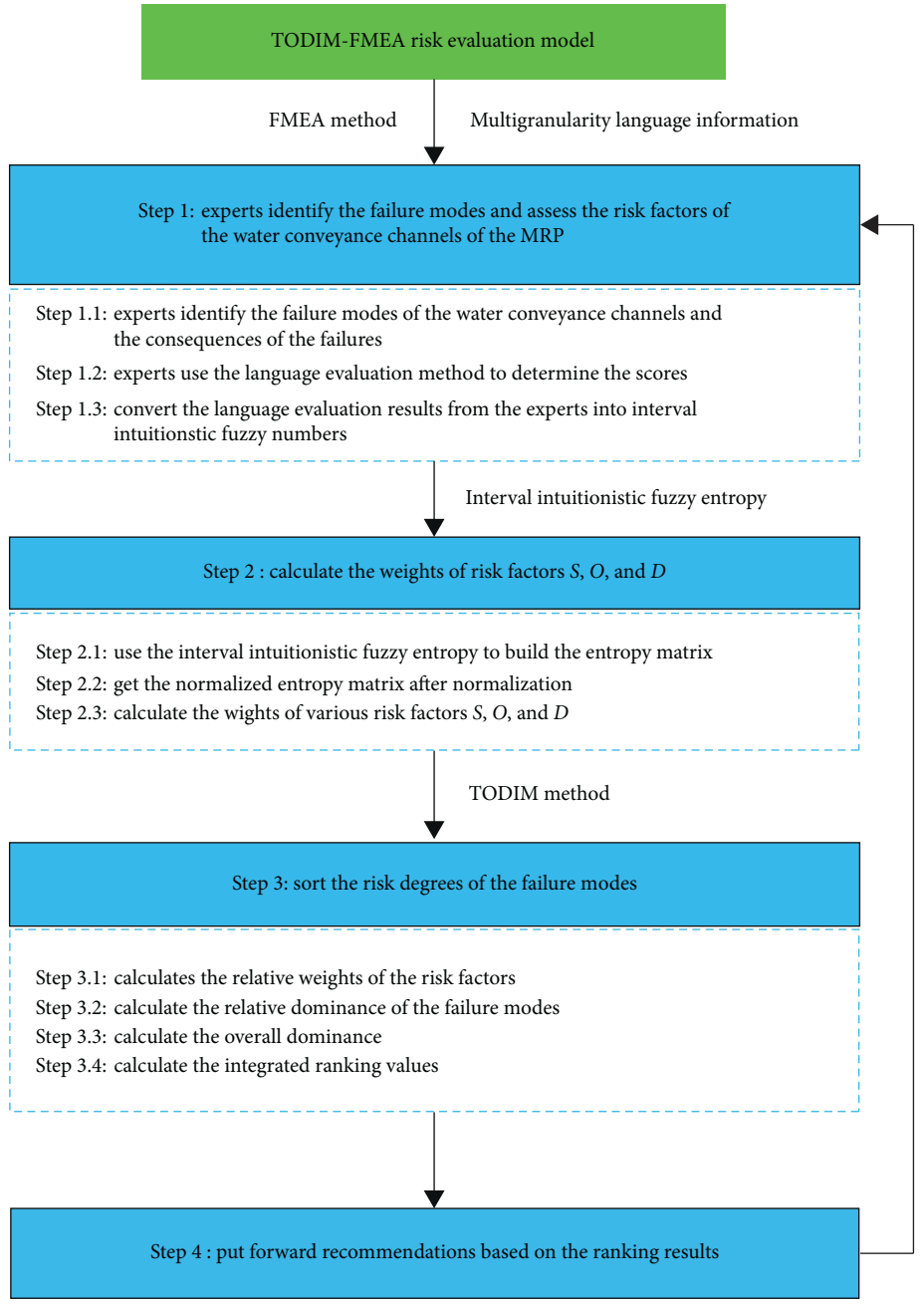


Figure 3: Risk assessment framework with TODIM-FMEA for the MRP.

$$\overline{E} = \begin{bmatrix} \overline{e}_{11} & \overline{e}_{12} & \cdots & \overline{e}_{1n} \\ \overline{e}_{21} & \overline{e}_{22} & \cdots & \overline{e}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{e}_{m1} & \overline{e}_{m2} & \cdots & \overline{e}_{mn} \end{bmatrix}, \tag{9}$$

where

$$\overline{e}_{ij} = \frac{e_{ij}}{\max\{e_1, e_2, \ldots, e_{mj}\}}, \quad (i = 1, 2, \ldots, m, j = 1, 2, \ldots, n). \tag{10}$$

Based on the standard entropy matrix $\overline{E}$, the weights of risk factors S, O, and D can be obtained through the following formula:

$$w_i = \frac{1 - \sum_{i=1}^{m} \overline{e}_{ij}}{\sum_{j=1}^{n} \left(1 - \sum_{i=1}^{m} \overline{e}_{ij}\right)}. \tag{11}$$

Step 3: the TODIM method is used to rank the failure modes.

The classical TODIM approach is extended to a multigranularity language environment for

calculation. The above steps have converted the multigranularity language information into interval intuitionistic fuzzy numbers, and then the integrated ranking of each failure mode is calculated according to the TODIM method. The detailed calculation steps are as follows.

(1) Calculate the relative weights of the risk factors:

The improved interval intuitionistic fuzzy entropy determined the weights of various indicators as $W = (w_1, w_2, \ldots, w_n)$ and the relative weights as $W_j' = (w_1', w_2', \ldots, w_n')$. The reference weights $w^* = \max\{w_1, w_2, \ldots, w_n\}$ can be determined based on the following formula:

$$w_j' = \frac{w_j}{w^*}. \tag{12}$$

(2) Calculate the relative dominance as formula (12):

$$\Phi_j(\mathrm{FM}_i, \mathrm{FM}_P) = \begin{cases} \sqrt{\dfrac{w_j'}{\sum_{j=1}^{n} w_j'} d(m_{ij}, m_{pj})}, & m_{ij} > m_{pj}, \\ 0, & m_{ij} = m_{pj}, \\ -(1/\theta)\sqrt{\dfrac{\sum_{j=1}^{n} w_j'}{w_j'} d(m_{ij}, m_{pj})}, & m_{ij} < m_{pj}, \end{cases} \tag{13}$$

where θ is the loss attenuation coefficient; $\theta > 1$ indicates that the influence of the loss will decrease, and $\theta < 1$ indicates that the influence of the loss will increase. $d(m_{ij}, m_{pj})$ indicates the distance between two interval intuitionistic fuzzy numbers, which is calculated with formula (5).

(3) Calculate the overall dominance and values of overall ranking:

The overall dominance can be determined from the following formula:

$$\delta(\mathrm{FM}_i, \mathrm{FM}_p) = \sum_{j=1}^{n} \Phi_j(\mathrm{FM}_i, \mathrm{FM}_p). \tag{14}$$

Values of overall ranking can be determined from the following formula:

$$\zeta_i = \frac{\sum_{k=1}^{m} \delta(\mathrm{FM}_i, \mathrm{FM}_p) - \min_i \sum_{k=1}^{m} \delta(\mathrm{FM}_i, \mathrm{FM}_p)}{\max_i \sum_{k=1}^{m} \delta(\mathrm{FM}_i, \mathrm{FM}_p) - \min_i \sum_{k=1}^{m} \delta(\mathrm{FM}_i, \mathrm{FM}_p)}. \tag{15}$$

(4) The failure modes are ranked according to the values of ζ_i.

4. Case Analysis

4.1. Case Description. This study takes the water channel of the Huixian section of the MRP as an example to analyze the operation risk. The Huixian section is located within Huixian City, Henan Province. The total length of the channel section is 48.951 km, with 43.631 km of the open channel and 5.320 km of the hydraulic structure. Huixian city has a warm temperate continental monsoon climate. In spring, it is windy and lacks rain; in summer, it is rainy and hot; in autumn, it is cool; and in winter, it is quite cold and lacks snow. The annual precipitation varies greatly. The average precipitation varies greatly depending on the month and season. While the average annual precipitation is 589.1 mm, the month of July sees nearly one-third of the total yearly precipitation, with an average of 182.3 mm. The foundation of the channel section contains both weak expansive soil and collapsible loess. The collapsible loess section of the foundation was treated by dynamic compaction during construction.

4.2. Processes of Operation Risk Assessment. This section attempts to validate the effectiveness of the proposed TODIM-FMEA model based on multigranularity language variables to assess the operation risks of the water channel in the MRP.

Step 1: firstly, the FMEA expert team comprises three experts who are engaged in the field of operation management for the South-to-North Water Diversion Project, and they are a scientific research expert, an engineering design expert, and an operation management expert. The expert weights are assigned to them according to their different knowledge structures and field experience, which are denoted as $W = (0.28, 0.31, 0.41)$. The experts selected three different language granularities, 5, 7, and 9, respectively, based on their own preference for carrying out the assessment. The experts assessed the six failure modes using three risk factors, i.e., severity (S), occurrence (O), and detection difficulty (D) with linguistic information, as shown in Table 2.

The multigranularity language evaluation sets are denoted as

$$\begin{aligned} E\text{: } S^5 &= \left\{s_{-2}^5, s_{-1}^5, s_0^5, s_1^5, s_2^5\right\}, \\ E\text{: } S^7 &= \left\{s_{-3}^7, s_{-2}^7, s_{-1}^7, s_0^7, s_1^7, s_2^7, s_3^7\right\}, \\ E\text{: } S^9 &= \left\{s_{-4}^9, s_{-3}^9, s_{-2}^9, s_{-1}^9, s_0^9, s_1^9, s_2^9, s_3^9, s_4^9\right\}. \end{aligned} \tag{16}$$

Scoring rules: scoring from the three different granular language variables, i.e., 5, 7, and 9, is listed as follows:

Table 2: Statistical table of the experts' evaluation information.

Experts	$E1$			$E2$			$E3$		
FMs	S	O	D	S	O	D	S	O	D
FM1	s_0^5	s_1^5	s_{-2}^5	s_0^7	s_2^7	s_{-2}^7	s_1^9	s_2^9	s_{-3}^9
FM2	s_1^5	s_{-1}^5	s_{-1}^5	s_1^7	s_{-2}^7	s_{-3}^7	s_2^9	s_{-2}^9	s_{-3}^9
FM3	s_2^5	s_{-2}^5	s_1^5	s_3^7	s_{-1}^7	s_{-3}^7	s_3^9	s_{-3}^9	s_{-1}^9
FM4	s_0^5	s_1^5	s_0^5	s_0^7	s_0^7	s_0^7	s_{-1}^9	s_3^9	s_1^9
FM5	s_1^5	s_0^5	s_1^5	s_2^7	s_{-1}^7	s_3^7	s_3^9	s_{-1}^9	s_2^9
FM6	s_1^5	s_{-1}^5	s_0^5	s_3^7	s_{-2}^7	s_2^7	s_2^9	s_{-1}^9	s_1^9

$$
\begin{aligned}
S^5 &= \left\{ s_{-2}^5 = \text{Lowest}, \ldots, s_0^5 = \text{Medium}, \ \ldots, s_2^5 = \text{Highest} \right\},\\
S^7 &= \left\{ s_{-3}^7 = \text{Lowest}, \ldots, s_0^7 = \text{Medium}, \ \ldots, s_3^7 = \text{Highest} \right\},\\
S^9 &= \left\{ s_{-4}^9 = \text{Lowest}, \ldots, s_0^9 = \text{Medium}, \ \ldots, s_4^9 = \text{Highest} \right\}.
\end{aligned} \tag{17}
$$

The language evaluation information is converted into the corresponding interval intuitionistic fuzzy number. And interval intuitionistic fuzzy evaluation matrix R is formed for the failure mode, as shown in Table 3. Then, according to Definition 7, each expert's evaluation information is integrated using the interval-valued intuitionistic fuzzy weighted arithmetic averaging (IIWAA) operator. Finally, the group integrated weight of interval-valued intuitionistic fuzzy judgment matrix $R' = [a_{ij}]_{m\times n}$ is obtained, as shown in Table 4.

Step 2: the weights are calculated based on interval intuitionistic fuzzy entropy. Firstly, the entropy value of the group integrated weight of interval-valued intuitionistic fuzzy judgment matrix is calculated. According to improved interval intuitionistic fuzzy entropy formula (7), the matrix is calculated as follows:

$$
E = \begin{bmatrix}
0.4581 & 0.2524 & 0.1142\\
0.2824 & 0.1742 & 0.1355\\
0.1884 & 0.1608 & 0.4776\\
0.3354 & 0.3000 & 0.4581\\
0.2261 & 0.3794 & 0.2322\\
0.2322 & 0.2420 & 0.3421
\end{bmatrix}. \tag{18}
$$

Then, the elements in the entropy matrix are normalized, and normalized entropy matrix $\overline{E}$ is built:

$$
\overline{E} = \begin{bmatrix}
1.0000 & 0.6652 & 0.2391\\
0.6165 & 0.4593 & 0.2837\\
0.4112 & 0.4239 & 1.0000\\
0.7321 & 0.7908 & 0.9592\\
0.4935 & 1.0000 & 0.4862\\
0.5069 & 0.6379 & 0.7163
\end{bmatrix}. \tag{19}
$$

Based on normalized entropy matrix $\overline{E}$, the weights of risk factors S, O, and D can be calculated through formula (11): $W_i = (0.3277, 0.3535, 0.3188)$.

Step 3: ranking failure modes of the water channels in the MRP with the TODIM method.

(1) According to formula (12), relative weights of the risk factors are calculated as $W_j' = (0.9272, 1.0000, 0.9018)$.

(2) Calculate the relative dominance.
As to the relative dominance, the decision maker's judgment is largely determined by the size of the loss attenuation coefficient θ, and θ is negatively correlated with the decision maker's degree of loss avoidance. When $\theta > 1$, it indicates that the impact of the loss will decrease and that the decision maker holds an attitude of risk aversion; when $\theta < 1$, it indicates that the impact of the loss will increase and that the decision maker holds an attitude of willingness to confront risks. If $\theta = 1$ is adopted, it indicates the decision maker holds a neutral attitude to risks. The distance is obtained through formula (5), and the relative dominance ϕ of the risk factors S, O, and D under different failure modes is determined through a pairwise comparison of the failure modes according to formula (13). The dominance matrix $\phi_c = [\phi_{ij}]_{m\times m}$ is generated, as shown in Table 5.

(3) Calculate the overall dominance matrix and integrated ranking values.
In accordance with formula (14), the overall dominance matrix $\delta = [\delta_{ij}]_{m\times m}$ can be derived, as shown in Table 6.

According to formula (15), the integrated ranking values of the risk levels of the six failure modes are $\zeta_1 = 0.0459$; $\zeta_2 = 0$; $\zeta_3 = 0.6035$; $\zeta_4 = 0.5227$; $\zeta_5 = 1$; $\zeta_6 = 0.7825$.

(4) Rank the failure modes.

Based on the overall dominance, the integrated sequence of ζ_i is ranked according to the values. The ranking result is $\zeta_5 > \zeta_6 > \zeta_3 > \zeta_4 > \zeta_1 > \zeta_2$. According to the ranking, abnormal leakage (FM6) poses the largest riesk, and the poor water delivery during ice periods poses the lowest level. The ranking result is shown in Figure 4.

According to the radar diagram in Figure 4, abnormal leakage is the most typical failure mode in the risk identification of the water channels in the MRP. Abnormal leakage will not only cause a loss of water quantity but, more seriously, will also cause erosion to the foundation and bank slopes, which will endanger the overall stability of the structures. Cracks and leakage will reduce material strength, thus accelerating crack development and leading to a vicious circle of deterioration. Therefore, abnormal leakage has an important effect on the safety of the water channels in the MRP. The failure mode of

Table 3: Interval-valued intuitionistic fuzzy judgment matrix.

Failure modes	Experts								
	*E*1			*E*2			*E*3		
	Risk factors								
	S	*O*	*D*	*S*	*O*	*D*	*S*	*O*	*D*
FM1	([0.4, 0.5], [0.4, 0.5])	([0.6325, 0.7071], [0.2254, 0.2929])	([0.064, 0.125], [0.784, 0.875])	([0.4286, 0.5], [0.4286, 0.5])	([0.7539, 0.7937], [0.1702, 0.2063])	([0.0787, 0.125], [0.8134, 0.875])	([0.6667, 0.707], [0.2546, 0.2929])	([0.7631, 0.7937], [0.1779, 0.2063])	([0.0390, 0.0625], [0.9047, 0.9375])
FM2	([0.6325, 0.7071], [0.2254, 0.2929])	([0.16, 0.25], [0.64, 0.75])	([0.16, 0.25], [0.64, 0.75])	([0.6546, 0.7071], [0.2441, 0.2929])	([0.0787, 0.125], [0.8134, 0.875])	([0.0337, 0.0625], [0.8934, 0.9375])	([0.7631, 0.7937], [0.1779, 0.2063])	([0.0878, 0.125], [0.8285, 0.875])	([0.0390, 0.0625], [0.9047, 0.9375])
FM3	([0.7368, 0.7937], [0.1566, 0.2063])	([0.064, 0.125], [0.784, 0.875])	([0.6325, 0.7071], [0.2254, 0.2929])	([0.8091, 0.8409], [0.13056, 0.1591])	([0.1837, 0.25], [0.6735, 0.75])	([0.0337, 0.0625], [0.8934, 0.9375])	([0.8165, 0.8409], [0.1367, 0.1591])	([0.0390, 0.0625], [0.9047, 0.9375])	([0.6667, 0.707], [0.2546, 0.2929])
FM4	([0.4, 0.5], [0.4, 0.5])	([0.6325, 0.7071], [0.2254, 0 .2929])	([0.4, 0.5], [0.4, 0.5])	([0.4286, 0.5], [0.4286, 0.5])	([0.4286, 0.5], [0.4286, 0.5])	([0.4286, 0.5], [0.4286, 0.5])	([0.1975, 0.25], [0.6914, 0.75])	([0.8165, 0.8409], [0.1367, 0.1591])	([0.6667, 0.707], [0.2546, 0.2929])
FM5	([0.6325, 0.7071], [0.2254, 0.2929])	([0.4, 0.5], [0.4, 0.5])	([0.6325, 0.7071], [0.2254, 0.2929])	([0.7539, 0.7937], [0.1702, 0.2063])	([0.1837, 0.25], [0.6735, 0.75])	([0.8091, 0.8409], [0.13056, 0.1591])	([0.8165, 0.8409], [0.1367, 0.1591])	([0.1975, 0.25], [0.6914, 0.75])	([0.7631, 0.7937, [0.1779, 0.2063])
FM6	([0.6325, 0.7071], [0.2254, 0 .2929])	([0.16, 0.25], [0.64, 0.75])	([0.4, 0.5], [0.4, 0.5])	([0.8091, 0.8409], [0.13056, 0.1591])	([0.0787, 0.125], [0.8134, 0.875])	([0.7539, 0.7937], [0.1702, 0.2063])	([0.7631, 0.7937], [0.1779, 0.2063])	([0.1975, 0.25], [0.6914, 0.75])	([0.6667, 0.707], [0.2546, 0.2929])

Note. FM: failure mode.

Table 4: Group integrated weight of interval-valued intuitionistic fuzzy judgment matrix.

Failure modes	Risk factors		
	S	O	D
FM1	([0.5356, 0.5984], [0.3396, 0.4016])	([0.7289, 0.7724], [0.1875, 0.2276])	([0.0585, 0.0999], [0.8409, 0.9001])
FM2	([0.6989, 0.7463], [0.2097, 0.2537])	([0.1059, 0.1250], [0.7663, 0.8380])	([0.0730, 0.1193], [0.8179, 0.8807])
FM3	([0.7945, 0.8289], [0.1400, 0.1711])	([0.0931, 0.1419], [0.7932, 0.8581])	([0.5235, 0.5798], [0.3631, 0.4201])
FM4	([0.3342, 0.4096], [0.5114, 0.5904])	([0.6830, 0.7308], [0.2241, 0.2692])	([0.5356, 0.5984], [0.3396, 0.4016])
FM5	([0.7559, 0.7954], [0.1683, 0.2046])	([0.2563, 0.3305], [0.5884, 0.6695])	([0.7495, 0.7900], [0.1727, 0.2100])
FM6	([0.7495, 0.7900], [0.1727, 0.2100])	([0.1516, 0.2133], [0.7116, 0.7867])	([0.6423, 0.6948], [0.2550, 0.3052])

Note. FM: failure mode.

Table 5: Relative dominance matrix of different failure modes of risk factors.

	Φ_S						Φ_O						Φ_D					
Failure modes	FM1	FM2	FM3	FM4	FM5	FM6	FM1	FM2	FM3	FM4	FM5	FM6	FM1	FM2	FM3	FM4	FM5	FM6
FM1	0	−0.2433	−0.3565	0.0325	−0.7958	−0.3178	0	0.1134	0.1121	0.0079	0.0822	0.1012	0	−0.0347	−0.7521	−0.7853	−1.0835	−0.9296
FM2	0.0261	0	−0.1409	0.0586	−0.0839	−0.0745	−0.9071	0	0.0061	−0.8468	−0.2809	−0.1130	0.0035	0	−0.7202	−0.7513	−1.0589	−0.1634
FM3	0.0383	0.0151	0	0.0738	0.0061	0.0071	−0.8974	−0.0487	0	−0.8340	−0.2839	−0.1118	0.0764	0.0732	0	−0.0350	−0.3482	−0.1848
FM4	−0.3025	−0.5457	−0.6866	0	−0.6297	−0.6203	−0.0634	0.1058	0.1042	0	0.0743	0.0933	0.0798	0.0763	0.0036	0	−0.3267	−0.1634
FM5	0.0855	0.0090	−0.0570	0.0676	0	0.0010	−0.6576	0.0351	0.0355	−0.5942	0	0.0215	0.1101	0.1076	0.0354	0.0332	0	0.0166
FM6	0.0341	0.0080	−0.0664	0.0666	−0.0094	0	−0.8101	0.0141	0.0140	−0.7467	−0.1721	0	0.0945	0.0166	0.0188	0.0166	−0.1634	0

Note. FM: failure mode.

Table 6: Overall dominance matrix.

δ	FM1	FM2	FM3	FM4	FM5	FM6
FM1	0	−0.1646	−0.9964	−0.7449	−1.7972	−1.1462
FM2	−0.8775	0	−0.8550	−1.5394	−1.4237	−0.3509
FM3	−0.7827	0.0396	0	−0.7953	−0.6260	−0.2895
FM4	−0.2861	−0.3636	−0.5788	0	−0.8821	−0.6903
FM5	−0.4620	0.1517	0.0139	−0.4934	0	0.0391
FM6	−0.6815	0.0387	−0.0336	−0.6635	−0.3449	0

Note. FM: failure mode.

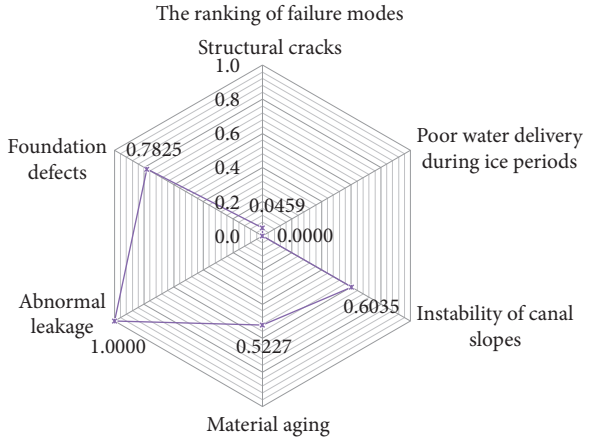


Figure 4: The ranking of six failure modes.

foundation defects poses the second largest threat. There is weak expansive soil in the Huixian section of the water channels in the MRP. Over the course of the operation, we should strengthen inspections of the geological risks and take engineering measures to reduce those risks in a timely manner when problems arise. Although the failure modes of structural cracks, instability of canal slopes, and material aging rank behind abnormal leakage and foundation defects, they are important failure modes as well for the safe operation of the channels in the MRP. Over the course of operation, some concrete measures should be put forward to improve and enhance the reliability of the water channels in the MRP. Poor water conveyance during ice periods poses the least risk because Huixian county is at a low latitude, and it will experience less harm to water delivery during ice periods in winter than northern regions will.

4.3. Comparison and Discussion. In this case, the two methods, i.e., interval intuitionistic fuzzy set TOPSIS and traditional RPN, were used to analyze risks, and the results were compared with those of the TODIM method, as shown in Table 7. The ranking results of the three methods are very similar, and the ranking of failure modes FM5 and FM2 is the same. That is, all methods ranked FM5 as the most serious failure mode and FM2 as the least serious failure mode, which shows that the three methods are all valid.

Table 7: Ranking comparison result.

Failure mode	RPN		TOPSIS		TODIM	
	RPN	Ranking	RCC	Ranking	δ	Ranking
FM1	117.056	4	0.6145	3	0.0459	5
FM2	71.58923	6	0.0022	6	0.0000	6
FM3	103.0053	5	0.3003	5	0.6035	3
FM4	245.9878	2	0.4135	4	0.5227	4
FM5	352.8724	1	0.9731	1	1.0000	1
FM6	222.4502	3	0.8903	2	0.7825	2

Note. FM: failure mode.

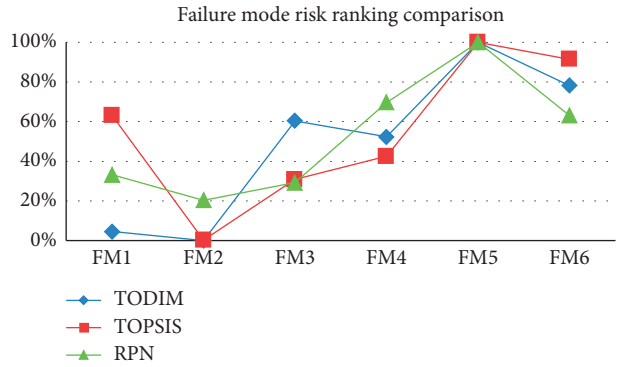


Figure 5: Failure mode risk ranking comparison.

While the comparison between the TODIM method and the TOPSIS method shows there is positional ranking crossing over for FM1 and FM3, the risk ranking of the remaining four failure modes is the same, which proves the effectiveness of the multigranularity language evaluation TODIM method.

Figure 5 depicts deviations of the failure modes with the most severe failure mode for each ranking method. In Figure 5, it can be seen that the fluctuation forms of the three methods are basically the same. The difference in risk identification results between TODIM and TOPSIS is due to the TODIM method taking into consideration the decision makers' psychological behavior and risk preference. Therefore, the multigranularity language TODIM method is more reliable than the traditional RPN method and the TOPSIS method.

The ranking of the improved FMEA risk evaluation approach differs from other approaches, and the reasons can be explained as follows:

(1) The calculation of the proposed model is based on multigranularity language variables. Experts can more accurately evaluate the FMEA failure modes with multigranularity language variables, which are converted into interval direct fuzzy numbers. The multigranularity language variables can quantitatively explain the evaluation results from experts.

(2) The traditional RPN calculation does not assign weights to risk factors S, O, and D. The proposed model adopted improved interval intuition entropy to assign weights to the risk factors, which can

determine the influence of risk factors in the evaluation process.

(3) The TODIM method proposed in this paper fully considers the psychological behaviors of experts, but the TOPSIS method assumes that experts are absolutely rational people. For the TODIM method, a pairwise comparison of all failure modes uses relative dominance to determine the overall dominance ranking. This causes the results to contain more information. It is found that the FMEA risk evaluation method based on multigranularity language TODIM can more accurately reflect the opinions of experts and more accurately obtain the relationship between various failure modes.

5. Conclusions

Channel engineering is one of the most important parts of system engineering in the MRP. This paper proposed a new FMEA risk evaluation model based on the multigranularity language information TODIM method. The multigranularity language, which can be converted into interval-valued intuition fuzzy sets, is used to evaluate the engineering failure modes. The degree of membership and the degree of nonmembership are used to express the decision maker's psychology of hesitation and quantify the risks. The weights of the risk factors, severity (S), occurrence (O), and detection difficulty (D), are calculated by using improved interval intuitionistic fuzzy entropy. The risk assessment model is built based on the TODIM-FMEA method. Finally, a case study of the Huixian section of the MRP is presented. The results show that abnormal leakage is the most risky failure mode. A comparison of three risk ranking methods shows that the proposed model is more effective than the traditional RPN method and the TOPSIS method.

The TODIM-FMEA risk evaluation model with multigranularity language is highly operational and quite practical, providing a new solution to the problem of operational safety risk evaluation in water conservancy projects. The limitation of this study is that the weights of experts were directly assigned, which may lead to a deviation in risk ranking results. Therefore, in the future, appropriate methods can be studied for calculating the weights of experts to optimize the risk assessment model.

Some suggestions should be put forward to control the operation risks of water channels in the MRP. From the perspective of engineering safety, once leakage is found, or the values of deformations and cracks prompt warnings from the structure monitoring system of the MRP, it is necessary to quickly and accurately find the source of the leakage to eliminate defects and damage from the leakage and to prevent defects or damage from expanding. From the perspective of water delivery, it is necessary to reduce the impact on dispatching and operation as much as possible. It is necessary to carry out underwater repair research to optimize underwater construction techniques.

Notations

FMEA:	Failure Mode and Effects Analysis
RPN:	Risk priority number
SNWDP:	The South-to-North Water Diversion Project
ERP:	The East Route Project
MRP:	The Middle Route Project
WRP:	The West Route Project
TOPSIS:	Technique for Order Preference by Similarity to Ideal Solution
S:	Severity
O:	Occurrence
D:	Detection difficulty
DEMATEL:	Decision-Making Trial and Evaluation Laboratory
TODIM:	Tomada de Decisão Interativa Multicritério
IIWAA:	The interval-valued intuitionistic fuzzy weighted arithmetic averaging.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge with gratitude the National Key R&D Program of China (no. 2018YFC0406905), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (no. 19YJC630078), Henan Overseas Expertise Introduction Center for Discipline Innovation: Smart Water, Youth Talents Teachers Scheme of Henan Province Universities (no. 2018GGJS080), the National Natural Science Foundation of China (nos. 71974056 and 71302191), the Foundation for Distinguished Young Talents in Higher Education of Henan (Humanities and Social Sciences), China (no. 2017-cxrc-023), the China Scholarship Council (no. 201908410388), and 2018 Henan Province Water Conservancy Science and Technology Project (GG201828). This study would not have been possible without their financial support.

References

[1] Office of the South-to-North Water Diversion Project Construction Committee, "The South-to-North Water Diversion Project," *Engineering*, vol. 2, no. 3, pp. 265–267, 2016.

[2] Z. Luo, S. Li, K. Hou, and G. Ji, "Spatial and seasonal bacterioplankton community dynamics in the main channel of the middle route of South-to-North Water diversion project," *Research in Microbiology*, vol. 170, no. 1, pp. 24–34, 2019.

[3] H. Shen, Q. Cai, and M. Zhang, "Spatial gradient and seasonal variation of trophic status in a large water supply reservoir for the South-to-North Water Diversion Project, China," *Journal of Freshwater Ecology*, vol. 30, no. 2, pp. 249–261, 2015.

[4] C. Tang, Y. Yi, Z. Yang, and X. Cheng, "Water pollution risk simulation and prediction in the main canal of the South-to-North Water Transfer Project," *Journal of Hydrology*, vol. 519, pp. 2111–2120, 2014.

[5] S. Li, J. Li, and Q. Zhang, "Water quality assessment in the rivers along the water conveyance system of the Middle Route of the South to North Water Transfer Project (China) using multivariate statistical techniques and receptor modeling," *Journal of Hazardous Materials*, vol. 195, pp. 306–317, 2011.

[6] X. Lei, H. Zheng, and L. Kong, "Emergency operation technologies for sudden water pollution accidents," *in Emergency operation technologies for sudden water pollution accidents in the middle route of south-to-north water diversion project*, pp. 1–8, L. Xiaohui, IntechOpen Publisher, London, UK, 2018.

[7] X. Du and L. Geng, "Risk analysis on operation of mid-route of South-to-North Water transfer project," *Water Resources and Hydropower Engineering*, vol. 42, no. 3, pp. 85–88, 2011.

[8] D. Hu, L. Zheng, S. Li, and G. Shi, "Study on the risk evaluation method for the open channel operation in the middle route of South-to-North Water diversion project," *South-to-north Water Transfer and Water Science and Technology*, vol. 11, no. 6, pp. 98–101, 2013.

[9] W. Cheng, J. Chen, and T. Liu, "Method for computing correlation coefficient between structural risks in water transfer project," *Journal of Wuhan University of Hydraulic and Electric Engineering*, no. 1, pp. 15–18, 2004.

[10] M. Zhou, J. Lian, X. Cheng, and X. Zhao, "Study on measures for main canal operation of middle route project of South-to-North Water diversion during ice period," *Yangtze River*, vol. 47, no. 21, p. 106, 2016.

[11] Y. Xiong, W. Qi, and Z. Wang, "Operation risk study on the middle route of the South-to North Water diversion project (part I) -risk identification in the middle route of the South-to-North water diversion project," *South-to-North Water Transfers and Water Science and Technology*, vol. 8, no. 3, pp. 1–5, 2010.

[12] V. R. Renjith, M. Jose kalathil, P. H. Kumar, and D. Madhavan, "Fuzzy FMECA (failure mode effect and criticality analysis) of LNG storage facility," *Journal of Loss Prevention in the Process Industries*, vol. 56, pp. 537–547, 2018.

[13] I. Lazakis, Y. Raptodimos, and T. Varelas, "Predicting ship machinery system condition through analytical reliability tools and artificial neural networks," *Ocean Engineering*, vol. 152, pp. 404–415, 2018.

[14] W. Choi, B. D. Youn, H. Oh, and N. H. Kim, "A Bayesian approach for a damage growth model using sporadically measured and heterogeneous on-site data from a steam turbine," *Reliability Engineering & System Safety*, vol. 184, pp. 137–150, 2019.

[15] K. Chang, "A more general risk assessment methodology using a soft set-based ranking technique," *Soft Computing*, vol. 18, no. 1, pp. 169–183, 2014.

[16] B. Vahdani, M. Salimi, and M. Charkhchian, "A new FMEA method by integrating fuzzy belief structure and TOPSIS to improve risk evaluation process," *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 1–4, pp. 357–368, 2015.

[17] R. Wang, J. Zhu, and Y. Li, "Improved FMEA risk assessment method based on intuitionistic fuzzy MULTIMOORA," *Computer Integrated Manufacturing System*, vol. 24, no. 2, pp. 290–301, 2018.

[18] I. Emovon, R. A. Norman, A. J. Murphy, and K. Pazouki, "An integrated multicriteria decision making methodology using compromise solution methods for prioritising risk of marine machinery systems," *Ocean Engineering*, vol. 105, pp. 92–103, 2015.

[19] J. Zhu, G. Li, R. Wang, and Y. Li, "FMEA based assessment of risk that metro train stops abnormally," *China Safety Science Journal*, vol. 27, no. 2, pp. 145–150, 2017.

[20] H.-C. Liu, J.-X. You, M.-M. Shan, and L.-N. Shao, "Failure mode and effects analysis using intuitionistic fuzzy hybrid TOPSIS approach," *Soft Computing*, vol. 19, no. 4, pp. 1085–1098, 2015.

[21] K.-H. Chang, Y.-C. Chang, and Y.-T. Lee, "Integrating TOPSIS and DEMATEL methods to rank the risk of failure of FMEA," *International Journal of Information Technology and Decision Making*, vol. 13, no. 06, pp. 1229–1257, 2014.

[22] L. Gomes and M. Lima, "TODIM: basics and application to multicriteria ranking of projects with environmental impacts," *Foundations of Computing and Decision Sciences*, vol. 16, no. 4, pp. 113–127, 1992.

[23] K. T. Atanassov, "Two theorems for intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 110, no. 2, pp. 267–269, 2000.

[24] E. Szmidt and J. Kacprzyk, "Distances between intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 114, no. 3, pp. 505–518, 2000.

[25] M. Delgado, F. Herrera, E. Herrera-Viedma, and L. Martinez, "Combining numerical and linguistic information in group decision making," *Information Sciences*, vol. 107, no. 1–4, pp. 177–194, 1998.

[26] Z. Xu and J. Chen, "Approach to group decision making based on interval-valued intuitionistic judgment matrices," *Systems Engineering-Theory & Practice*, vol. 27, no. 4, pp. 126–133, 2007.

Hindawi
Complexity
Volume 2020, Article ID 8854462, 8 pages
https://doi.org/10.1155/2020/8854462

Research Article

Evaluation Index System for Agricultural Water Management in Targeted Poverty Alleviation Based on 3E Model

Yingfeng Chen,[1] **Shuyang Zhu**,[2] **and Ming Fan**[1]

[1]*School of Management, Jiangsu University, Zhenjiang 212013, China*
[2]*Jiangsu University of Technology, Changzhou 213001, China*

Correspondence should be addressed to Shuyang Zhu; zhushuyang@jsut.edu.cn

Received 26 August 2020; Revised 9 October 2020; Accepted 23 October 2020; Published 1 December 2020

Academic Editor: Rui Wang

Agricultural water management provides the basic support and guarantee for targeted poverty alleviation. This paper presents a 3E + 1 evaluation model for the performance of agricultural water management in targeted poverty alleviation based on 3E theory, which is more scientific, reasonable, and reliable. On this basis, an evaluation index system including three levels of indicators is designed, and the weight of each evaluation index and performance evaluation model is determined. A case study of a county in the old district of Maoshan in Jiangsu province was conducted using the proposed evaluation theory and methods. The results show that the overall performance of agricultural water management for targeted poverty alleviation in this area was good but needs more improvement in innovation. The evaluation results are roughly consistent with the evaluations of higher authorities, experts, and scholars, which proves that the evaluation system is scientific and reasonable.

1. Introduction

In the process of targeted poverty alleviation, agricultural water management is essential and indispensable for ensuring rural water safety, improving rural water environment, enhancing agricultural comprehensive productivity, and improving farmers' production and living standards [1]. It is crucial for winning the tough battle against poverty. Since the 18th National Congress of the Communist Party of China, targeted poverty alleviation work has been guided by Xi Jinping's thought on socialism with Chinese characteristics for a new era. China has been deepening the implementation of the water management policy of "prioritizing water conservation, balanced space, and systemic governance." Adhering to the basic strategy of targeted poverty alleviation and poverty reduction, the Chinese government has put an emphasis on severely impoverished areas and optimized the supply policy by promoting the construction and management of agricultural water conservancy in poor areas. By solidly promoting industry poverty alleviation, designated poverty alleviation, counterpart support, and "five in one" water conservancy poverty alleviation work in the old rural areas, China has been working to win the battle against poverty and build a moderately prosperous society in all respects [2].

Despite the great achievements in agricultural water management for targeted poverty alleviation, the work needs to be scientifically and effectively evaluated. In this way, problems can be identified through evaluation, thus providing a basis for poverty alleviation policy making in the post-poverty alleviation era. For this purpose, it is necessary to design a scientific and reasonable performance evaluation system through quantitative methods. By digging into the problems of agricultural water management for targeted poverty alleviation, the evaluation system can promote the high-quality development of the work and help to establish a long-term agricultural water management mechanism for targeted poverty alleviation in the post-poverty alleviation era. This has become a key scientific frontier issue that needs to be systematically studied and thoroughly discussed.

2. Method of Constructing Indicator System

2.1. Construction of a Performance Evaluation Model for Agricultural Water Management in Targeted Poverty

Alleviation Based on 3E Model. The 3E theory (economy, efficiency, effectiveness) proposed by Professor Checkland (Checkland, PB) based on SSM (soft system methodology) from a system perspective has been regarded as the basic model for public policy evaluation and has become the basis for various late-stage policy evaluations. SSM is a methodology for recognizing and handling complex problems because when soft factors such as politics, society, culture, and human behaviors are mixed into the system, traditional hard system analysis (e.g., modeling profit maximization) often loses its advantages and sometimes fails [3].

The object of performance evaluation is a poverty alleviation method that implements accurate identification, targeted assistance, and management [4], and the evaluation content is the quality and effect of agricultural water management according to targeted poverty alleviation policy [5]. In building a performance evaluation system, it is critically important to decide how to effectively measure whether the allocation and management of resources is reasonable and whether poverty alleviation initiatives are effective as well as the quality of poverty alleviation, outcome satisfaction, and sustainability of targeted poverty alleviation policies [6].

A performance evaluation framework based on 3E theory can ensure the scientific validity of the performance evaluation of agricultural water targeted poverty alleviation. Therefore, according to the basic paradigm of policy evaluation, this paper details the performance evaluation process into five stages: policy formulation and goal setting, targeted poverty alleviation input, alleviation process, alleviation output, and alleviation outcome, which fully integrates 3E theory into the whole process to extract the evaluation dimension [7]. Throughout the performance evaluation, targetedness is stressed and is also the essential difference between targeted poverty alleviation and wide-reaching poverty alleviation. Therefore, targetedness needs to be fully reflected in each dimension of the evaluation model [8]. In addition, this paper innovatively incorporates the sustainability of agricultural water management in targeted poverty alleviation into the targeted poverty alleviation evaluation system [9]. This can effectively reflect the sustainability and stability of agricultural water management in targeted poverty alleviation, forming a 3E + 1 evaluation model (as shown in Figure 1).

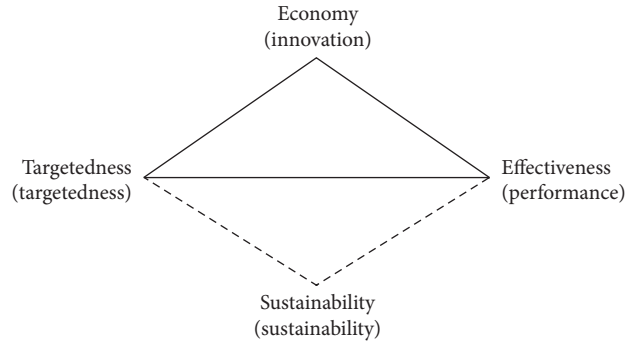


Figure 1: The $3E+1$ evaluation model of the performance of agricultural water management in targeted poverty alleviation.

2.2. Construction of the Index System for Agricultural Water Management in Targeted Poverty Alleviation. The construction of the performance evaluation index system is a multiobjective and multilevel complex systematic engineering, which should follow the principles of comprehensiveness, scientificity, comparability, operability, and sustainability [10]. The performance evaluation indicators should reflect not only the main contents of the assessment of effectiveness but also the targetedness, stability, and quality of the targeted poverty alleviation work so that the overall evaluation function of the indicator system is greater than the simple accumulation of indicators. It is necessary to build a clear and reasonable hierarchical systematic structure [11] and avoid overlapping between the indicators while retaining mutual connections [12]. The evaluation indicators established should be universally applicable and feasible [13] and comparable and dynamic for comparisons between different regions, time, and space [14]. Evaluation indicators need to be feasible so the data to be used should be drawn from existing data sources and be verifiable [15]; mutual inclusion and implicit relationship should be avoided; work involved in the evaluation system should also be long-term and continuous [16]. Based on the goals, characteristics, and related literature research of agricultural water management in targeted poverty alleviation [17], the indicator system is designed as Table 1.

2.3. Weight Determination of the Performance Evaluation Index System for Agricultural Water Management in Targeted Poverty Alleviation. This study adopts analytic hierarchy process (AHP) to evaluate and rank the four primary indicators of the performance index system for agricultural water management in targeted poverty alleviation, while the average weighting method is used for the secondary and tertiary indicators. Analytic hierarchy process (AHP), first proposed by Professor Saaty in the 1970s, is a structured decision-making method that combines qualitative and quantitative analyses and is applicable to the analysis of multiple indicator systems [18].

2.3.1. Construct a Judgment Matrix Based on the Existing Evaluation System. First, the indicators at the criterion level in the performance evaluation indicators are analyzed, using an AHP analysis model on a scale from 1 to 9. The indicators at the criterion level in the performance evaluation indicators are compared, and a judgment matrix at the four levels of performance, targetedness, innovation, and sustainability is obtained.

From the statistical results, we can conclude that the relative importance values of sustainability to targetedness, performance, and innovation are 3, 5, and 7, respectively; the relative importance values of targetedness to performance and innovation are 3 and 6, respectively; the relative importance value of performance to innovation is 4; the other two relative importance values can be inferred by analogy from the above results. This leads to the results in Table 2.

Table 1: Performance evaluation index system of agricultural water management in targeted poverty alleviation.

Target	Criterion	Dimension	Index	Index properties	Index definition	Assignment method
Agricultural water management in targeted poverty alleviation performance	Performance	Drinking water safety	Local access to safe drinking water rate	Quantitative	The ratio of population having access to safe drinking water to the total population of the area	$b_{11} = X_{\text{population having access to safe drinking water}}/Y_{\text{the total population of the area}}$
		Agricultural water infrastructure construction	Rural centralized water supply rate	Quantitative	Rural centralized water supply reflects the improvement of access to safe drinking water	$b_{21} = X_{\text{number of villages with centralized water supply}}/Y_{\text{the total number of villages}}$
			Effective agricultural irrigation rate	Quantitative	The ratio of the irrigated area of the local farmland to the cultivated area reflects the extent of agricultural irrigation	$b_{22} = X_{\text{number of irrigated area of local farmland}}/Y_{\text{number of cultivated area}}$ $b_{21} = X_{\text{water supply village}}/Y_{\text{total village}}$
			Agricultural water-saving probability	Quantitative	The ratio of the water-saving irrigated area to the irrigated area on the farmland reflects the extent of water-saving irrigation	$b_{23} = X_{\text{number of irrigated water-saving area}}/Y_{\text{number of irrigated area on farmland}}$
			Reinforcement rate of dangerous reservoirs	Quantitative	Reinforced dangerous reservoirs to total dangerous reservoirs ratio reflects poverty alleviation by flood control and disaster mitigation projects	$b_{24} = X_{\text{number of reinforced dangerous reservoirs}}/Y_{\text{the total number of dangerous reservoirs}}$
			Rural hydropower efficiency expansion rate	Quantitative	Rural small hydropower project implementation reflects the hydropower project poverty alleviation	$b_{25} = X_{\text{number of new installed kilowatt-hours}}/Y_{\text{number of installed kilowatt-hours}}$
		Reservoir resettlement	Reservoir resettlement rate	Quantitative	The ratio of the number of resettled reservoirs to the total number of reservoirs	$b_{31} = X_{\text{number of resettled reservoirs}}/Y_{\text{the total number of reservoirs}}$
		Social recognition	Agricultural water management in poverty alleviation rate	Quantitative	The ratio of the number of people lifted out of poverty by agricultural water management in poverty alleviation to the number of local people lifted out of poverty	$b_{41} = X_{\text{number of people lifted out of poverty by agriculture water}}$
			Public acceptance	Quantitative	The ratio of the number of people contented with agricultural water management in poverty alleviation to the total number of surveyed people	$b_{42} = X_{\text{number of people contented with agricultural water management in targeted poverty alleviation}}/Y_{\text{the total number of surveyed people}}$
	Targetedness	Survey on needs for poverty alleviation	Pairing officials with low-income people	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
			Targeted poverty alleviation in the agricultural water sector	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
		Agricultural water management in targeted poverty alleviation	Forming an implementation plan for agricultural water targeted poverty alleviation	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
		Use of funds for agricultural water management in poverty alleviation	Follow-up audit rate of agricultural water management in targeted poverty alleviation projects	Quantitative	The ratio of the number of agricultural water management projects in targeted poverty alleviation tracked and audited to the total number of agricultural water management projects implemented	$b_{71} = X_{\text{number of agricultural water management projects tracked and audited}}/Y_{\text{the total number of agricultural water management projects in poverty alleviation implemented}}$

Table 1: Continued.

Target	Criterion	Dimension	Index	Index properties	Index definition	Assignment method
	Innovation	Agricultural water management in targeted poverty alleviation	Local media reports on poverty alleviation	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
		Agricultural water management training	Local agricultural water management training rate	Quantitative	The ratio of the number of trainees to the number of people in the industry reflects personnel development in this sector	$b_{91} = X_{\text{number of trainees}}/Y_{\text{number of people in the industry}}$
	Sustainability	Scientific and effective assessment	Evaluation of incentives for agricultural water-relieving officials	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
		Agricultural water project reserves	Implementation status of agricultural water project reserves	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
		Ecological environment	Implementation of the river system	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1
			Comprehensive control rate of water loss and soil erosion	Quantitative	Soil and water loss management reflects the importance attached to environmental improvement	$b_{122} = X_{\text{number of treated water loss and soil erosion areas}}/Y_{\text{number of water loss and soil erosion areas}}$
			River regulation rate	Quantitative	River channel improvement raises flood control standards	$b_{123} = X_{\text{length of regulated river}}/Y_{\text{the full length of the river}}$
		Agricultural water management practices in targeted poverty alleviation	Discipline and law violations in agricultural water management in targeted poverty alleviation	Qualitative	Very satisfied, satisfied, neutral, dissatisfied, very dissatisfied	5, 4, 3, 2, 1

Table 2: Weights of the four factors at the criterion level.

Index	Sustainability	Targetedness	Performance	Innovation
Sustainability	1	3	5	7
Targetedness	0.333333	1	3	6
Performance	0.200000	0.333	1	4
Innovation	0.142857	0.166667	0.250000	1

2.3.2. Calculate the Weight Vector. First, calculate each column in the judgment matrix to normalize

$$W_j = \frac{w_{ij}}{\sum_{i=1}^{n} w_{ij}}. \quad (1)$$

Second, calculate the canonical column average (i.e., the vector of weights sought)

$$w_j^* = \frac{\sum_{i=1}^{n} w_j}{n}. \quad (2)$$

Finally, calculate the maximum eigenvalue

$$\lambda_{\max} = \frac{1}{n} \sum_{1}^{n} \frac{(BW)_i}{w_i}. \quad (3)$$

After column normalization, normalizing the average, calculating the maximum eigenvalue, and consistency test, we obtained the weight values between the four factors at the criterion level, and the specific results and calculation process are shown in Table 3.

According to the results of the comparison judgment matrix, after column normalization and normalized averaging using hierarchical analysis (AHP), we obtained the weights of the four indicators of performance, targetedness, innovation, and sustainability. Then, we need to test whether the weight values of the four indicators are acceptable by the consistency test. According to the judgment matrix, the eigenvectors and eigenvalues of the judgment matrix can be obtained: the maximum eigenvalue is 4.182997, and the result of the consistency test $cr = 0.068538 < 0.1$. At this time, it is considered that the weight given by the criterion level in the performance evaluation index system is acceptable. The results of the weights are 0.548172 for sustainability, 0.269686 for targetedness, 0.130931 for performance, and 0.051212 for innovation [19].

2.4. Construction of Performance Evaluation Model for Agricultural Water Management in Targeted Poverty Alleviation. Considering various characteristic factors in evaluating the performance of agricultural water management in targeted poverty alleviation, this study constructed a model for evaluating the performance, as shown in equation (4).

$$Z = \sum_{i=1}^{4} Q_i \left(\sum_{j=1}^{n} Q_{ij} \left(\sum_{k=1}^{t} Q_{ijk} V_{ijk} \right) \right). \quad (4)$$

In this model, z represents the final score of the performance of agricultural water management in targeted poverty alleviation; Q_i represents the weight of the first-level indicators, Q_{ij} represents the weight of second-level indicators under each first-level indicator; n represents the number of second-level indicators under each first-level indicator; Q_{ijk} represents the weight of the third-level indicators under the second-level indicators; t represents the number of the third-level indicators under the second-level indicators; and V_{ijk} represents the score value of each third-level indicator.

In order to accurately retain the weight scores of the final indicators of performance evaluation, results of equation (4) are multiplied by 1000 to set the score interval of performance evaluation at (0–1000).

2.5. Performance Evaluation Standards for Agricultural Water Management in Targeted Poverty Alleviation. Since the final performance scores are by multiplied by 1000 in the above evaluation model, the performance evaluation of agricultural water management in targeted poverty alleviation adopts a scoring system ranging from 0 to 1000, which is set to four levels: excellent (>900), good (800–900, including 800), moderate (600–800, including 600), and poor (<600).

3. Application of Performance Evaluation for Agricultural Water Management in Targeted Poverty Alleviation in a County in the Old District of Maoshan in Jiangsu

Theoretical basis and performance evaluation system only address the concept and operational methods of performance evaluation of agricultural water management in targeted poverty alleviation from a theoretical point of view. To test whether the method is practical and operable, it should also be applied to the case study. Therefore, a county in the old district of Maoshan in Jiangsu province is chosen as an example to comprehensively evaluate the performance of agricultural water management in targeted poverty alleviation by applying the performance evaluation theory and index system proposed in this paper.

3.1. Overview of the County in Case Study. Located in the old district of Maoshan in the southwest of Jiangsu province, the county covers an area of 1,535 square kilometers, including 1.12 million mu of the cultivated land, 328,000 mu of the forest land, and 426,000 mu of rivers and lakes. There are many types of landforms such as low mountains, hills, and plains in the area. The southern, western, and northern sections are higher, and the intermediate, central, and eastern sections are flatter. The south is a low area with steeper mountains; the northwest is a hilly area with rolling hills; the intermediate and central areas are flat from west to east and

Table 3: Weights of the four factors at the criterion level.

Index	Sustainability	Targetedness	Performance	Innovation
Sustainability	1	3	5	7
Targetedness	0.333333	1	3	6
Performance	0.200000	0.333333	1	4
Innovation	0.142857	0.166667	0.250000	1
		Column normalization		
	0.596591	0.666667	0.540541	0.388889
	0.198864	0.222222	0.324324	0.333333
	0.119318	0.074074	0.108108	0.222222
	0.085227	0.037037	0.027027	0.055556
		Norm column average		
	0.548172	0.269686	0.130931	0.051212
		Maximum eigenvalue		
	1.081032	1.068347	1.022120	1.011498
		4.182997		
		Consistency check		
	Consistency index CI		**0.060999**	
	Average random one-time indicator CR		**0.068538**	

are a plain polder area. With subtropical monsoon climate, the county sees four distinct seasons, abundant rainfall, and long frost-free period. The average annual temperature is 17.5°C; the average annual precipitation is 1149.7 mm; the annual frost-free period is 250 days. There is an average of 1992.5 hours of sunlight per year with prevailing wind blowing from the east. It belongs to the Taihu Lake water system, which is located in the west water network area of Taihu Lake with criss-crossing river networks and scattered reservoirs and ponds. There are 426,000 acres of water area and 2 large reservoirs with a storage capacity of more than 100 million cubic meters. The water quality has always maintained the drinking water standard of National Level II.

In 2019, there were 60,200 qualified registered poor households and 11,300 low-income farmers in the county. Among the qualified registered poor households, there are three main types, namely, the average poor households, households enjoying the minimum living guarantee, and households enjoying the five guarantees. Most of these poor households are from households enjoying the minimum living guarantee: 0.44 million, accounting for 70.9% of the total. There are 0.14 million average poor households and 0.04 million households enjoying the five guarantees, accounting for 22.9% and 6.2% of the total, respectively. Statistics show that there are many causes of poverty for low-income farmers in the county, such as disease, disability, school, disaster, lack of land technology funds, and limitations on their own development. However, the two main factors that contribute to poverty are illness and disability. There are 0.29 million (47%) people who are ill or have a medical condition and 0.17 million (28.07%) with disabilities, which together account for 75.1% of the total number of low-income rural households.

3.2. The Application of the Performance Evaluation of Agricultural Water Management in Targeted Poverty Alleviation

3.2.1. Agricultural Water Performance Evaluation Process. According to the process and steps of performance evaluation, we visited the Agricultural Water Department, Poverty Alleviation Office, and other departments, selected some poor towns and villages in the county for on-site investigation and verification by random sampling, and investigated and visited 57 poor villages. We used seminars, questionnaires, and other investigation methods comprehensively during the on-the-spot investigation to understand the implementation of the responsibility system of agricultural water management in targeted poverty alleviation, the implementation of agricultural water projects, the pairing assistance of agricultural water management officials, and the use of agricultural water funds.

3.2.2. Sources of Data. The data for indicators are mainly from the 2019 National Poverty Alleviation Information System, the county's poverty alleviation office, the county's agricultural water management department, questionnaires, and surveys.

3.2.3. The Content of Performance Evaluation of Agricultural Water Management in Targeted Poverty Alleviation. According to the previous evaluation theory and evaluation method, the performance of agricultural water management in targeted poverty alleviation in this county is scored as shown in Table 4.

According to the evaluation results, the county's overall performance of agricultural water management in targeted poverty alleviation is excellent. We can see that the highest score is 100% (for targetedness), and the score for sustainability is 95.45%, for performance is 88.72% and for innovation is 52.94%. The results indicate that in the work of targeted poverty alleviation, the county's agricultural water sector has done a better job in the targetedness and meticulousness of the work, but there are still shortcomings in innovations. This has a certain relationship with the nature of work in agricultural water conservancy, because most of the projects of agricultural water management are water conservancy projects which feature procedural and normative work. Therefore, the region should pay attention to

Table 4: Performance scores of agricultural water management in targeted poverty alleviation in a county in the old district of Maoshan in Jiangsu.

Criterion	Dimension	Index	Standard score	Score (points)
Performance (A)	Drinking water safety (A_1)	Local access to safe drinking water rate (A_{11})	33	33
	Agricultural water infrastructure construction (A_2)	Rural centralized water supply rate (A_{21})	7	7
		Effective agricultural irrigation rate (A_{22})	7	7
		Agricultural water-saving pipes probability (A_{23})	7	3
		Reinforcement rate of dangerous reservoirs (A_{24})	7	5
		Rural hydropower efficiency expansion rate (A_{25})	7	5
	Reservoir resettlement (A_3)	Reservoir resettlement rate (A_{31})	33	30
	Social recognition (A_4)	Poverty alleviation rate (A_{41})	16	16
		Mass recognition (A_{42})	16	12
Targetedness (B)	Surveys on the need for agricultural water management in targeted poverty alleviation (B_1)	Pairing situation between agricultural water management officials and low-income people (B_{11})	45	45
		Targeted poverty alleviation by agricultural water management department (B_{12})	45	45
	Agricultural water management in targeted poverty alleviation (B_2)	Forming an implementation plan for agricultural water targeted poverty alleviation (B_{21})	90	90
	Use of the funds for agricultural water management in targeted poverty alleviation (B_3)	Follow-up audit rate of agricultural water management projects in targeted poverty alleviation (B_{31})	90	90
Innovation (C)	Agricultural water management in targeted poverty alleviation (C_1)	Local media reports on agricultural water management in poverty alleviation (C_{11})	17	9
	Agricultural water management personnel training (C_2)	Local agricultural water management personnel training rate (C_{21})	17	8
	Scientific and effective assessment (C_3)	Evaluation of incentives among officials of agricultural water management in poverty alleviation (C_{31})	17	10
Sustainability (D)	Agricultural water management project reserves (D_1)	Implementation status of agricultural water management project reserves (D_{11})	183	164
	Ecological environment (D_2)	Implementation of the river system (D_{21})	61	61
		Comprehensive treatment rate of soil erosion (D_{22})	61	61
		River regulation rate (D_{23})	61	55
	Practices in agricultural water management in targeted poverty alleviation (D_3)	Discipline and law violations in agricultural water management in targeted poverty alleviation (D_{31})	183	183
Performance evaluation score of agricultural water management in targeted poverty alleviation			1000	939

the standardization of agricultural water management in targeted poverty alleviation work and strengthen the innovation of targeted poverty alleviation work in the field of agricultural water conservancy so as to improve the quality and efficiency of targeted poverty alleviation work.

4. Conclusion

Based on 3E theory, this paper builds a 3E + 1 performance evaluation model of agricultural water management in targeted poverty alleviation, which improves the scientificity, rationality, and reliability of the performance evaluation. On this basis, a performance evaluation index system was designed, the corresponding index weights were determined, and the evaluation model was constructed. Based on this, a case study was applied to a county in the old district of Maoshan, Jiangsu. By analyzing the development of agricultural water management in targeted poverty alleviation work in the county in 2019, it is concluded that the performance evaluation results of agricultural water management in targeted poverty alleviation are basically consistent with the performance analyses of higher-level departments, relevant experts, and scholars, which proves that the theory and method proposed in this paper are scientific and applicable. However, in the practice of performance evaluation, it is still necessary to appropriately revise the evaluation method according to the characteristics and development changes of the evaluation object in order to obtain more reasonable and reliable evaluation results.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Major Project of Philosophy and Social Science Research in Universities of Jiangsu Provincial Department of Education in 2020.

References

[1] Z. C. Hui, "Give full play to the supporting role of the water conservancy industry and help to take the lead in winning the tough battle against poverty," in *Procedings of the Annual Conference of the Chinese Water Conservancy Society*, pp. 2–205, Sichuan, China, October 2016.

[2] P. B. Checkland, "Towards a system-based methodology for real-word problem solving," *Journal of system Engineering*, vol. 3, pp. 87–116, 1972.

[3] P. B. Checkland, "Towards a system-based methodology for real-word problem solving," *Journal of Applied System Engineering*, vol. 3, pp. 87–116, 1972.

[4] S. G. Wang and Z. H. Guo, "On China's targeted poverty alleviation," *Guizhou Social Sciences*, vol. 5, pp. 147–150, 2015.

[5] W. L. Jiang, "Research on the characteristics of precise poverty alleviation in water conservancy and its performance evaluation system," in *Proceedings of Annual Conference of the Chinese Water Conservancy Society*, pp. 244–247, Sichuan, China, October 2016.

[6] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[7] L. Sun, *Research on Performance Evaluation of Poverty Alleviation Project*, China Agricultural University, Beijing, China, 2015.

[8] Y. Fu and Y. R. Zhang, "Performance evaluation of poverty alleviation and development in Lanzhou city and its enlightenment," *Journal of Hunan Agricultural University (Social Science Edition)*, vol. 12, no. 5, pp. 25–30, 2011.

[9] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[10] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: a comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

[11] C. X. Liu, H. F. Zhao, and L. X. Wei, "An empirical study on the performance evaluation index system of Guangxi village-level targeted poverty alleviation," *Forestry Economics*, vol. 10, p. 31, 2018.

[12] Y. Dai, "Construction and application of the index system for the performance evaluation of targeted poverty alleviation work," *World of Labor Security*, vol. 5, p. 23, 2017.

[13] S. P. Hu, "Research on the construction of the index system for performance evaluation of precise poverty alleviation," *Journal of Shenyang Agricultural University (Social Science Edition)*, vol. 18, no. 5, pp. 513–520, 2016.

[14] C. G. Zhang, "On the construction of an indicator system for the comprehensive construction of a well-off society in rural areas," *Journal of Theoretical Sciences*, vol. 1, pp. 32-33, 2006.

[15] Q. Zhang and Z. L. Shi, "Construction of performance evaluation index system for poverty alleviation in my country," *Education and Economics*, vol. 4, p. 38, 2018.

[16] A. X. Chen and Y. Liu, "Research on the performance evaluation of the implementation of targeted poverty alleviation in my country by AHP," *Journal of Huaqiao University (Philosophy and Social Sciences Edition)*, vol. 1, pp. 116–129, 2017.

[17] W. Lei, C. Zhang, and L. Liu, "Research on Guizhou province water conservancy targeted poverty alleviation implementation evaluation index system," *Journal of Irrigation and Drainage*, vol. 12, pp. 127–130, 2018.

[18] X. Yang, "Research on performance evaluation of poverty alleviation projects from a precise perspective," *Financial Economy*, vol. 2, pp. 23–25, 2017.

[19] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

Hindawi
Complexity
Volume 2020, Article ID 8853735, 14 pages
https://doi.org/10.1155/2020/8853735

Research Article

Multisystem Optimization for an Integrated Production Scheduling with Resource Saving Problem in Textile Printing and Dyeing

Haiping Ma,[1] **Chao Sun**,[1] **Jinglin Wang**,[2] **Zhile Yang**,[3] **and Huiyu Zhou**[4]

[1]*Department of Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang 312000, China*
[2]*Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China, Ningbo, Zhejiang 315000, China*
[3]*Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China*
[4]*Department of Informatics, University of Leicester, Leicester, LE1 7RH, UK*

Correspondence should be addressed to Haiping Ma; mhping1981@126.com

Received 15 September 2020; Revised 15 October 2020; Accepted 26 October 2020; Published 18 November 2020

Academic Editor: Shi Cheng

Resource saving has become an integral aspect of manufacturing in industry 4.0. This paper proposes a multisystem optimization (MSO) algorithm, inspired by implicit parallelism of heuristic methods, to solve an integrated production scheduling with resource saving problem in textile printing and dyeing. First, a real-world integrated production scheduling with resource saving is formulated as a multisystem optimization problem. Then, the MSO algorithm is proposed to solve multisystem optimization problems that consist of several coupled subsystems, and each of the subsystems may contain multiple objectives and multiple constraints. The proposed MSO algorithm is composed of within-subsystem evolution and cross-subsystem migration operators, and the former is to optimize each subsystem by excellent evolution operators and the later is to complete information sharing between multiple subsystems, to accelerate the global optimization of the whole system. Performance is tested on a set of multisystem benchmark functions and compared with improved NSGA-II and multiobjective multifactorial evolutionary algorithm (MO-MFEA). Simulation results show that the MSO algorithm is better than compared algorithms for the benchmark functions studied in this paper. Finally, the MSO algorithm is successfully applied to the proposed integrated production scheduling with resource saving problem, and the results show that MSO is a promising algorithm for the studied problem.

1. Introduction

Industry 4.0, a new wave of industrialization, driven by smart information and communication technology, sparks a transformative view of framing process manufacturing and factory management practice [1–3]. With the recent technical advances in industry, such as sensor network, cloud computing, and artificial intelligence, production manufacturing is becoming hyperconnected because of the communication and control between more components and parts than those in the past [4–6]. So, the optimization and decision for manufacturing is generally impossible with analytical mathematical tools or traditional optimization methods [7, 8]. Meanwhile, manufacturing involves multiple subsystems, with multiple inputs, multiple outputs, multiple objectives, and multiple constraints. Its optimization and decision cannot be simply treated as a classic multiobjective optimization problem. We treat such manufacturing as a complex multisystem because it is not merely a single optimization problem but is rather a combination of several multiobjective optimization problems [9, 10]. That is, manufacturing is composed of multiple subprocesses, and each subprocess can be considered as an independent subsystem. These subsystems are coupled with the others, and they share some objectives or constraints with each other, and others are unique to a particular subsystem but must still be satisfied. Therefore, the optimization and decision for manufacturing in industry 4.0 is a more challenging problem than traditional multiobjective optimization in a single system.

In recent years, there are a few studies on the optimization of complex multisystem problems. Martins and

Lambe [11] presented the multidisciplinary design optimization (MDO) method, which was a framework of optimization methods for solving complex systems that involved multiple disciplines or subsystems. The basic principle of MDO was that the solution of a complex system depended not only on the solution of individual subsystems but also on their interactions, and its disadvantage was that it was only a framework that provided conceptual structures, but it did not specify the details of the underlying algorithms within that framework. That is, depending on the specific complex problem or the designer's preference, MDO required the designer to select particular optimization algorithms as the kernel components of MDO, which itself was a difficult problem. For the optimization of the complex multisystem, many researchers preferred to translate it as a multiobjective optimization problem and then adopted classical heuristic methods to seek optimal solutions. Zhang et al. [12, 13] proposed an efficient evolutionary optimization framework based on decomposition, which decomposed a complex problem into a series of single-objective optimization subproblems and then used the interactive information between adjacent subproblems to accelerate the optimization of each subproblem to obtain the optimal solution. Antonio and Coello [14] proposed a coevolutionary optimization (CO) method, inspired by biological coevolution and divided-and-autonomous strategy, and it established multiple competing or cooperating populations for evolving multiple subsystems to achieve global optimization. Ong and Gupta [15, 16] proposed a multiobjective multifactorial evolutionary algorithm (MO-MFEA) for solving multiple tasks in a complex system. Its basic idea was to unify the solution space of different tasks into an extended solution space and then performed implicit genetic transfer operation, which could exploit the transferable knowledge between multiple optimization tasks. The simulation results showed that MO-MFEA was competitive on multitask benchmarks. Recently, Du and Simon [17, 18] proposed a new optimization framework called BBO/Complex for solving the complex system using biogeography-based optimization (BBO). Its advantage was that it combined multipopulation architecture with the low-level optimization strategy to obtain a single method. The experimental results showed that the proposed BBO/Complex successfully solved four real-world optimization problems.

The aim of this paper is to propose and study a multisystem optimization (MSO) algorithm, combining the multipopulation architecture of BBO/Complex with the classical NSGA-II to establish within-subsystem evolution and cross-subsystem migration operators and then apply the new MSO algorithm to solving the production scheduling with resource saving problem in textile printing and dyeing plants. This paper shows how the multipopulation architecture can be integrated with popular heuristic methods to extend the application scenario of optimization algorithms and then presents a comparative study on multisystem benchmark functions and a textile printing and dyeing scheduling problem. The method in this paper could also serve as a template for the extension of any other heuristic methods to multisystem optimization.

The motivation of proposing MSO algorithm in this research is twofold. First, we have observed that there widely exist a large number of complex coupled systems in manufacturing, as mentioned above, and their optimization is more challenging and difficult than traditional multiobjective optimization in a single system. Second, we have observed that two main foundations of the MSO algorithm, multipopulation and NSGA-II, have proven themselves to be effective optimization architecture and method. Combining these two observations leads us to propose multisystem optimization, as a high-performing optimization method.

The original contributions of this paper include the following. (a) A new real-world-based textile printing and dyeing production scheduling problem considering resource saving is formulated as a multisystem optimization problem. (b) The idea of multisystem optimization is proposed in the heuristic computing field to establish the new optimization paradigm. Results show that the MSO algorithm outperforms compared algorithms for most of benchmark functions that we study. (c) The MSO algorithm solves the textile printing and dyeing production scheduling problem considering resource saving.

The remainder of this paper is organized as follows. Section 2 builds a mathematical model of textile printing and dyeing production scheduling problem considering resource saving. Section 3 realizes the MSO algorithm integrating multipopulation and NSGA-II, and Section 4 verifies its superiority with simulation results. Section 5 applies the MSO algorithm to solve the proposed production scheduling problem in textile printing and dyeing. Some concluding remarks and directions for future work are provided in Section 6.

2. Problem Descriptions and Modeling

The multisystem optimization problem studied in this paper is derived from a textile printing and dyeing plant. Its main processes include dyeing, printing, and finishing, and each process consists of multiple procedures. For example, the dyeing process includes singeing, bleaching, mercerizing, setting, and so on. According to raw materials and fabric structures, the process flow is not a completely fixed model for specific printing and dyeing requirements. Meanwhile, with the rapid development of manufacturing, resource saving and environmental protection have been increasingly concerned. Textile printing and dyeing is an energy- and resource-intensive manufacturing process and uses physical and chemical methods to make products out of the raw textile, with extensive electricity and water as the treatment mediums. Therefore, it is urgent for manufacturers to improve production efficiency while reducing electricity consumption and water wastage.

In this section, we propose an integrated optimization model for textile printing and dyeing by considering both production efficiency and resource saving. Figure 1 shows a workflow of textile printing and dyeing processes, where the model includes a production scheduling subsystem based on production orders and a resource saving subsystem based on

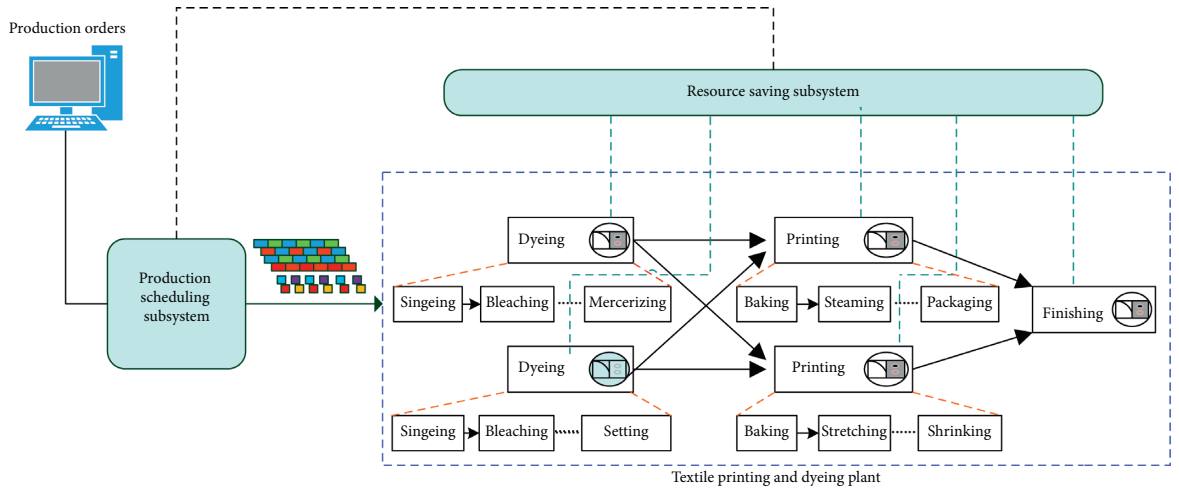


Figure 1: Workflow of textile printing and dyeing.

environment protection standard. These two subsystems are coupled with the other and share partial variables and constraints. The whole model is a hybrid flow-shop scheduling problem with unrelated parallel machines, and it is a nondeterministic polynomial hard (NP-hard) problem. In the following, we will focus on a mathematical formulation of this integrated optimization model. For convenience, the symbols and notations used in the model are shown in Table 1.

For a production scheduling subsystem in textile printing and dyeing, its mathematic model is formulated as a multiobjective optimization problem. Suppose the subsystem schedules a set of jobs $\{J_i\}_{i=1}^{n}$, and each job is completed orderly by m procedures. Each procedure has M_m unrelated parallel machines, and each job is worked on one machine in a procedure. A set-up time is required before a procedure of a job is processed on a machine. The production scheduling subsystem has two conflicting objectives of makespan and production cost, which should be minimized, respectively. This subsystem is defined as follows:

$$\min F_1 = (f_{11}, f_{12}), \tag{1}$$

$$f_{11} = \max_{i\in 1,2,\ldots,n} \{D_i\}, \tag{2}$$

$$f_{12} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} C_{ijk}. \tag{3}$$

In equation (1), F_1 denotes the production scheduling subsystem, including makespan f_{11} and production cost f_{12}. In equation (2), D_i is the finishing time of the ith job through all procedures, which is iteratively calculated by

$$\begin{aligned} D_{ij} &= \max_{j\in 1,2,\ldots,m-1} \left\{D_{i(j-1)} + T_{ijk}, D_{(i-1)(j+1)}\right\}, \\ D_i &= D_{i(m-1)} + T_{imk}, \end{aligned} \tag{4}$$

where T_{ijk} is the processing time of the jth procedure of the ith job running on the kth machine and D_{ij} is the finishing time of the ith job through the jth procedure. In equation (3), C_{ijk} is production cost of the jth procedure of the ith job running on the kth machine. If the jth procedure of the ith job is processed by the kth machine, then $w_{ijk} = 1$; otherwise, $w_{ijk} = 0$.

For a resource saving subsystem in textile printing and dyeing, electricity consumption, water wastage, and processing cost caused by these components are the most important factors, which vary with the process sequences. The mathematic model of this subsystem is also formulated as a multiobjective optimization problem, including three conflicting objectives of electricity consumption, water wastage, and processing cost, which should be minimized, respectively. The subsystem is defined as follows:

$$\min F_2 = (f_{21}, f_{22}, f_{23}), \tag{5}$$

$$\begin{aligned} f_{21} = {} & \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} P_{ijk}^{\mathrm{PE}} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{SE}} P_{jk}^{\mathrm{SE}} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{BE}} P_{jk}^{\mathrm{BE}} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk}\left(T_{ijk} + T_{ijk}^{\mathrm{SE}} + T_{ijk}^{\mathrm{BE}}\right) P_{jk}^{\mathrm{OE}}, \end{aligned} \tag{6}$$

$$f_{22} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} Q_{ijk}^{\mathrm{PW}} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{SE}} Q_{jk}^{\mathrm{SW}}, \tag{7}$$

$$f_{23} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} \kappa_{ijk}. \tag{8}$$

In equation (5), F_2 denotes the resource saving subsystem, including electricity consumption f_{21}, water wastage f_{22}, and the corresponding processing cost f_{23}. Electricity consumption denoted by equation (6) is composed of four parts: processing electricity consumption, set-up electricity consumption, standby electricity consumption, and auxiliary electricity consumption. The first term in equation (6) represents processing

Table 1: The symbols and notations used in the proposed model.

Indices	
I	Job
J	Procedure
k	Machine
n	Number of jobs
m	Number of procedures
M_m	Number of parallel machines
Variables	
w_{ijk}	= 1 if the jth procedure of the ith job is processed on the kth machine
$\{J_i\}_{i=1}^{n}$	Job sequence $\{J_1, J_2, J_3, \ldots, J_n\}$
Parameters	
C_{ijk}	Production cost of the jth procedure of the ith job on the kth machine
κ_{ijk}	Unit processing cost of the jth procedure of the ith job on the kth machine
D_i	Finishing time of the ith job through all procedures
D_{ij}	Finishing time of the ith job through the jth procedures
T_{ijk}	Processing time of the jth procedure of the ith job on the kth machine
T_{ijk}^{SE}	Set-up time of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
T_{ijk}^{BE}	Standby time of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{ijk}^{PE}	Unit processing power of the jth procedure of the ith job on the jth machine
P_{jk}^{SE}	Unit set-up power of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{jk}^{BE}	Unit standby power of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{jk}^{OE}	Auxiliary unit power of the kth machine during the jth procedure
Q_{ijk}^{PW}	Unit water wastage of the kth machine when processing the jth procedure of the ith job
Q_{jk}^{SW}	Unit water wastage of the kth machine in the set-up stage of the jth procedure

electricity consumption, where P_{ijk}^{PE} is the unit processing power of the kth machine when processing the jth procedure of the ith job. The second term represents set-up electricity consumption, where T_{ijk}^{SE} and P_{jk}^{SE} are, respectively, the set-up time and unit power of the kth machine when the $(j-1)$th procedure changes to the jth procedure for the ith job. The third term represents standby electricity consumption, where T_{ijk}^{BE} and P_{jk}^{BE} are, respectively, the standby time and unit power of the kth machine when the $(j-1)$th procedure changes to the jth procedure for the ith job. The fourth term represents auxiliary electricity consumption of the auxiliary equipments in the machining process, which is relevant to the production time and auxiliary power, with the former consisting of processing time, set-up time, and standby time, and P_{jk}^{OE} is the auxiliary unit power of the kth machine in the jth procedure.

Equation (7) denotes water wastage, which is different from electricity consumption. It considers two types of water wastage, including processing water wastage and set-up water wastage, and water wastage for auxiliary equipments and the standby stage of machines is neglected. In equation (7), the first term represents processing water wastage, and Q_{ijk}^{PW} is the unit water wastage of the kth machine when processing the jth procedure of the ith job. The second term represents set-up water wastage, and Q_{jk}^{SW} is the unit water wastage of the kth machine in the set-up stage of the jth procedure.

Equation (8) denotes the total processing cost, including raw material cost, electricity consumption cost, water consumption cost, and so on. κ_{ijk} is unit processing cost of the jth procedure of the ith job running on the kth machine.

Furthermore, for the proposed integrated optimization model of textile printing and dyeing, the solutions must satisfy the following constraints.

Each machine handles exactly one procedure in a job:

$$\sum_{i=1}^{n} w_{ijk} = 1, \quad \text{for } j \in \{1, 2, 3, \ldots, m\}, k \in \{1, 2, 3, \ldots, M_m\}, \tag{9}$$

and each procedure of each job is processed in only one machine sometime:

$$\sum_{k=1}^{M_m} w_{ijk} = 1, \quad \text{for } i \in \{1, 2, 3, \ldots, n\}, j \in \{1, 2, 3, \ldots, m\}. \tag{10}$$

In the above constraints, equation (9) is stand for assignment of one procedure in only one job to a machine, and equation (10) is stand for assignment of each procedure of each job to only one machine.

Now, an integrated optimization model for textile printing and dyeing has been presented, and the following section develops a multisystem optimization algorithm that will be used to solve the proposed optimization problem.

3. Multisystem Optimization

This section first presents a MSO framework, which can serve as a template for extending any other heuristic methods to the MSO algorithm. Then, it presents the implementation of the proposed MSO algorithm.

3.1. MSO Framework. A MSO problem in this paper consists of multiple subsystems, which are coupled with the others and share partial objectives or constraints. That is, each subsystem in a MSO problem has not only coupled objectives and constraints but also independent objectives and constraints. Therefore, multisystem optimization is more complicate than traditional multiobjective optimization.

Suppose that we have a complex system that consists of several subsystems. Without loss of generality, all subsystems are assumed to be minimization problems. Inspired by implicit parallelism of multipopulation heuristic approaches [19, 20], we first treat a subpopulation as a subsystem to optimize a subsystem optimization problem by excellent evolution operators. Then, we realize information sharing between multiple subsystems by migration, based on the relations of sharing variables and similarity levels between objectives and constraints, to accelerate the global optimization of the whole system. Based on the above idea, each subsystem is comprised of three sets of elements. The first set includes candidate solutions to the subsystem optimization problem. The second and third sets include the objectives and constraints of the subsystem. The MSO algorithm mainly includes two steps: evolution within subsystems and sharing information via migration across subsystems. We refer to these two types of operators as within-subsystem evolution and cross-subsystem migration. The MSO framework is illustrated in Figure 2, where a complex multisystem problem includes multiple coupled subsystems, and each subsystem includes multiple objectives and constraints. Within-subsystem evolution is used in each subsystem, and cross-subsystem migration is used between multiple subsystems.

3.2. Implementation of MSO Algorithm. In the proposed MSO architecture, we use a modified version of NSGA-II [21], initially designed for single systems with multiobjectives, as a within-subsystem evolution operator. The modified NSGA-II employs solution ranks as selection probabilities considering the relative performance of a candidate solution because each subsystem has its own set of candidate solutions, objectives, and constraints, and the ranks assigned to the candidate solutions in a subsystem denote the relative fitness of those solutions only in that particular subsystem. Then, we recombine the candidate solutions using any desired recombination method in heuristic methods. Finally, we mutate the child population and replace the parents with the children.

Cross-subsystem migration is an important operator in the MSO algorithm. For the development of heuristic methods including the proposed MSO algorithm in this paper, we must consider two challenges. One is to converge to the optimal solutions. To address this challenge in the MSO algorithm, we define similarity levels for both objectives and constraints. If two subsystems have high similarity levels, the optimization problems of those subsystems are similar to each other. This also means that the features that are important in one subsystem have a similar level of importance in the other subsystem. Migration between subsystems with similar objectives and constraints is expected to be helpful for all such subsystems. Another important challenge is to maintain population diversity as the main factor that enables the population to improve. If the population has a low diversity, most of candidate solutions are similar to each other, and the probability that a candidate solution improves after migration is low. In this case, migration may not effectively contribute to improvement in the population.

In cross-subsystem migration, we first use the similarity levels of both constraints and objectives to find pairs of subsystems that are suitable for migration. The similarity level calculation is based on the fast similarity level calculation (FSLC) [10] shown in Algorithm 1, where G and H are the sets of objective costs or constraints of two solutions in different subsystems. The pair probability P_{sub} between the subsystems is calculated as follows:

$$P_{\text{sub}} = \frac{1}{2}\left(\frac{\text{SL}_1}{\text{SL}_{1,\max}} + \frac{\text{SL}_2}{\text{SL}_{2,\max}}\right), \tag{11}$$

where SL_1 and SL_2 are, respectively, the objective and constraint similarity levels between two solutions in different subsystems and $\text{SL}_{1,\max}$ and $\text{SL}_{2,\max}$ are, respectively, the maximum objective and constraint similarity levels in the entire system.

Then, we use Euclidian distances between candidate solutions in a pair of subsystems to decide which solutions migrate between each other. In a multisystem problem, subsystems typically have different candidate solution structures, and the solution variables are at least partly different between subsystems. Table 2 illustrates an example about candidate solution structures in different subsystems. In the table, the candidate solution in subsystem 1 has four variables composed of Var-1, Var-2, Var-4, and Var-5, and the candidate solution in subsystem 2 has three variables composed of Var-1, Var-3, and Var-5. Apparently, these two candidate solutions have two variables Var-1 and Var-5 in common. To calculate their distances, we need to have identical solution structures and use an N/A value to denote a missing variable, shown as [1, 2, N/A, 0, 3] for the solution in subsystem 1 and [3, N/A, 4, N/A, 5] for the solution in subsystem 2.

Next, we use the partial distance strategy (PDS) [17, 22] to calculate Euclidian distance between candidate solutions in different subsystems, which is given as follows:

$$\sigma_{ikjl} = \begin{cases} \dfrac{N}{I_{ikjl}} \sqrt{\displaystyle\sum_{s=1}^{N} \left(y_{ik}(s) - y_{jl}(s)\right)^2 I_{ikjl}(s)}, & \text{if } I_{ikjl} > 0, 0, \text{if } I_{ikjl} = 0, \end{cases} \tag{12}$$

where

$$\begin{aligned} I_{ikjl} &= \sum_{s=1}^{N} I_{ikjl}(s), \\ I_{ikjl}(s) &= \begin{cases} 0, & \text{if } y_{ik}(s) = N/A \text{ or } y_{jl}(s) = N/A, \\ 1, & \text{if } y_{ik}(s) \neq N/A \text{ and } y_{jl}(s) \neq N/A. \end{cases} \end{aligned} \tag{13}$$

In equation (12), σ_{ikjl} denotes the partial distance between candidate solution y_{ik} in subsystem i and candidate

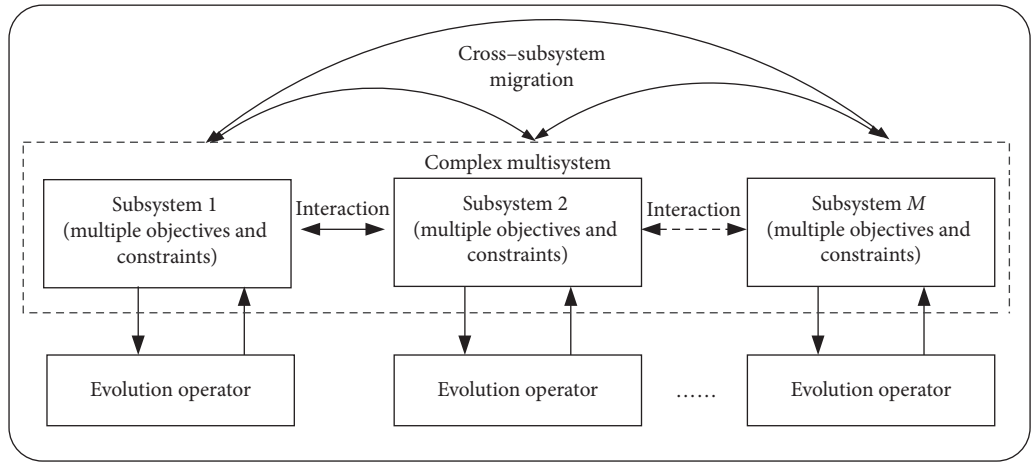


Figure 2: The MSO framework, mainly including within-subsystem evolution and cross-subsystem migration.

```
Set the objective or constraint similarity level SL = 0
For each g ∈ G, where G is the set of objectives or constraints of one solution
   For each h ∈ H, where H is the set of objectives or constraints of another solution
      If g = h, then
         SL = SL + 1
      End if
   End for
End for
```

Algorithm 1: Similarity level calculation across subsystems.

solution y_{jl} in subsystem j, which quantifies the difference level between each pair of solutions. N denotes the total number of solution variable for the entire system, and $y_{ik}(s)$ denotes the sth solution variable of y_{ik}.

Finally, we perform migration between the immigrating solution y_{ik} and the emigrating solution y_{jl} cross the subsystems. We use partial distance σ to calculate migration probabilities and then use the roulette-wheel selection method to probabilistically choose emigrating solution based on the migration probabilities. Migration is defined as

$$y_{ik}(s) \leftarrow y_{jl}(s). \tag{14}$$

Equation (14) states that a solution variable in the emigrating solution replaces one in the immigrating solution. In cross-subsystem migration, each solution variable in each immigrating solution in a subsystem has a chance to be replaced by a solution variable of an emigrating solution from another subsystem. One generation of the proposed MSO algorithm is shown in Algorithm 2.

4. Simulation Results

This section presents the optimization performance of the proposed MSO algorithm on the benchmark functions from the literature [23]. These functions are originally to evaluate the performance of MO-MFEA. Here, we take a task of a function as a subsystem, and each subsystem includes a multiobjective function. Multitask benchmark functions naturally change to multisystem functions suitable for the evaluation of the proposed MSO. These functions are built considering intersecting degrees of solution variables: complete intersection, partial intersection, and no intersection, and similarity degree of objectives: high similarity, medium similarity, and low similarity. Accordingly, there are nine combinations in total, and they are briefly summarized in Table 3, where the functions are named as follows: the first capital letters "*C*," "*P*," and "*N*" denote "complete," "partial," and "no," respectively, the second capital letter "I" denotes "intersection", the third capital letters "*H*," "*M*," and "*L*" denote "high," "medium," and "low," respectively, and the last capital letter "*S*" denotes "similarity". The aim of such categories is to have a comprehensive evaluation of the proposed method. In addition, $sim(T_1, T_2)$ denotes the similarity coefficient between subsystems T_1 and T_2 in a benchmark function.

In this experiment, we use the proposed MSO algorithm for solving simultaneously two subsystems of a benchmark function. To show the benefits of the proposed method, the results are compared with those of MO-MFEA and improved NSGA-II [24]. MO-MFEA is a multifactorial evolutionary algorithm [15], which is to simultaneously tackle multiple optimization tasks. Improved NSGA-II uses multiple populations with different crossover operators to separately solve each subsystem in a benchmark function. To ensure a fair comparison, the parameter configurations of

Table 2: Candidate solution structures in different subsystems.

Subsystem 1	Var-1	Var-2	Var-3	Var-4	Var-5
Solution	1	2	N/A	0	3
Subsystem 2	Var-1	Var-2	Var-3	Var-4	Var-5
Solution	3	N/A	4	N/A	5

these algorithms are adjusted to be optimal through many experiments to obtain the appropriate performance. The population size of NSGA-II is set to 100 for solving a single subsystem, while the population size of the proposed MSO algorithm and MO-MFEA is set to 200 for solving a benchmark function with two subsystems. The maximal number of function evaluations on a subsystem is set to 100,000 for NSGA-II, while the proposed MSO algorithm and MO-MFEA use 200,000 for a benchmark function since they solve two subsystems in a function together at a time. That is, for each subsystem, their maximal number of function evaluations is still 100,000. For all algorithms, we use the same simulated binary crossover with a crossover probability $p_c = 1$ and use the same polynomial mutation with a mutation probability $p_m = 1$. To compare with existing results, we use the inverted generational distance (IGD) in the literature [23] to evaluate the performance on each subsystem of the considered benchmark function. A small IGD value means both good convergence and good diversity for a minimization problem. Table 4 shows the average and standard deviation of IGD values for the proposed MSO algorithm, MO-MFEA, and NSGA-II, and all results are computed from 30 independent simulations.

The results shown in Table 4 lead to the following discussion. First, considering a similarity degree of objectives, we take CIHS, PIHS, and NIHS as a group of high similarity, CIMS, PIMS, and PIHS as a group of medium similarity, and CILS, PILS, and NILS as a group of low similarity. From Table 4, we observe that for high and medium similarity groups, the proposed MSO algorithm performs better than MO-MFEA and NSGA-II for all benchmark functions. For the low similarity group, the proposed MSO algorithm performs better than MO-MFEA and NSGA-II but worse than MO-MFEA for two of the benchmark functions (CILS and NILS). It indicates that objective similarity has a great influence on optimization performance of the proposed MSO algorithm, and it is an important factor composing cross-subsystem migration in the MSO algorithm.

Second, considering intersection degree of solution variables, we take CIHS, CIMS, and CILS as a group of complete intersection, PIHS, PIMS, and PILS as a group of partial intersection, and NIHS, NIMS, and NILS as a group of no intersection. From Table 4, we observe that for partial intersection, the proposed MSO algorithm is better than MO-MFEA and NSGA-II for all benchmark functions. For the complete intersection group and no intersection group, the proposed MSO algorithm is better than NSGA-II but worse than MO-MFEA for CILS and NILS. It indicates that solution variable intersection has a certain effect on optimization performance of the proposed MSO algorithm, and it is another important factor composing cross-subsystem migration in the MSO algorithm.

Furthermore, we also use symbol "†" to show statistically significant differences between the proposed MSO algorithm and compared algorithms based on the p value, which is smaller than 0.05 regarded as the significance level. In Table 4, out of 18 groups of data, there are 14 statistically significant differences between the proposed MSO algorithm and MO-MFEA and there are 16 statistically significant differences between the proposed MSO algorithm and NSGA-II. Based on this result, the probability that MSO and compared algorithms are from the same distribution is low. It indicates that the proposed MSO is an independent algorithm.

Combining with the above observations, we find that as a whole, the proposed MSO algorithm performs better than the other algorithms used in this paper for the most of benchmark functions. It implies that the MSO algorithm can fully consider the inheritance of evolution information and relationship between optimization environment and performance, and migration between subsystems can effectively utilize these factors to accelerate global optimization.

5. Application of MSO to Manufacturing Optimization

In this section, we use the proposed MSO algorithm to solve the integrated production scheduling with resource saving in textile printing and dyeing described in Section 2. In the traditional textile printing and dyeing industry, production scheduling is conducted by workers. Manual scheduling is very labor-consuming, and it usually leads to unsatisfactory schemes. Moreover, resource consumption is rarely considered when workers schedule jobs. In this experiment, 10 instances with different procedures and number of machines are investigated, which are shown in Table 5. These instances are generated from a real textile dyeing and printing plant located in Eastern China, and we use uniform distribution for the range of model parameters based on manual scheduling experience of workers because their real values are difficult to obtain. The job processing time in each machine is uniformly distributed in [50 100] minutes, and the set-up time and standby time of a machine are uniformly distributed in [5 10] minutes. Production cost and unit processing cost are uniformly distributed in [50 120] and [20 80], respectively, and various unit electricity consumption and unit water wastage are uniformly distributed in [20 50] and [2 10], respectively. The tuning parameters of the proposed MSO algorithm and the compared algorithms are the same as those used in the benchmark functions. In addition, the proposed integrated optimization model is a combinatorial optimization problem. We use permutation coding for each procedure, and the decoding includes the sorting of jobs and the allocation of machines. For job

Calculate the similarity levels of objectives and constraints between each pair of subsystems
Calculate the rank of each solution y_{ik} in subsystem i
Calculate migration probability λ_{ik}, which is linearly proportional to the rank of y_{ik}
For each subsystem i
  Perform within-subsystem evolution using the modified NSGA-II
    Find a suitable subsystem j to pair with subsystem i based on similarity levels of objectives and constraints
  For each solution y_{ik}
      Calculate partial distances $\{\sigma_{ikjl}\}$ between y_{ik} and each solution in subsystem j
    For each solution variable s
      Use λik to decide whether or not to immigrate to y_{ik}
      If migrating, then
          Select emigrating solution y_{jl} using roulette wheel selection, and its probability is proportional to partial distances $\{\sigma_{ikjl}\}$
        y_{ik} (s) $\longleftarrow$ y_{jl} (s)
      End if
    End for
  End for
End for

Algorithm 2: One generation of the proposed MSO algorithm.

Table 3: Benchmark function sets.

Function	$Sim\ (T_1, T_2)$	Subsystem no.	Pareto set	Pareto front	Properties
CIHS	0.97	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Separable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	Concave, unimodal Separable
CIMS	0.52	T_1	$x_1 \in [0,1]$, $x_i = 1, i = 2{:}\ 10$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{10})^T = s_{cm2}$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Nonseparable
CILS	0.07	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Separable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$	Convex, multimodal Nonseparable
PIHS	0.99	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$	Convex, unimodal Separable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{ph2}$	$f_2 = 1 - \sqrt{f_1}$ $0 \leq f_1 \leq 1$	Convex, multimodal Separable
PIMS	0.55	T_1	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{pm1}$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable
PILS	0.002	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{pl2}$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
NIHS	0.94	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$ $0 \leq f_1 \leq 1$	Convex, unimodal Separable
NIMS	0.51	T_1	$x_1 \in [0,1], x_2 \in [0,1]$, $\sum_{i=1}^{3} f_1^2 = 1$	$x_i = 1, i = 3{:}\ 20$ $f_i \geq 0, i = 1, 2, 3$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1], x_2 \in [0,1]$, $x_i = 0, i = 3{:}\ 20$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, unimodal Nonseparable
NILS	0.001	T_1	$x_1 \in [0,1], x_2 \in [0,1]$, $(x_3, \ldots, x_{25})^T = s_{nl1}$	$\sum_{i=1}^{3} f_1^2 = 1$ $f_i \geq 0, i = 1, 2, 3$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1], x_2 \in [0,1]$, $x_i = 0, i = 3{:}\ 50$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable

Table 4: The average and standard deviation of IGD values obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II. The best result in each row is shown in bold font.

Function	$Sim\ (T_1, T_2)$	Subsystem no.	IGD		
			MSO	MO-MFEA	NSGA-II
CIHS	0.97	T_1	**7.51*E*-05 ± 8.63*E*-06**	4.16*E*-04 ± 2.67*E*-05†	2.17*E*-03 ± 4.32*E*-04†
		T_2	**9.25*E*-05 ± 1.31*E*-05**	5.64*E*-04 ± 3.28*E*-05†	5.31*E*-03 ± 1.66*E*-04†
CIMS	0.52	T_1	**7.36*E*-03 ± 5.14*E*-04**	1.19*E*-02 ± 4.32*E*-02†	8.01*E*-01 ± 3.42*E*-02†
		T_2	**3.42*E*-03 ± 7.10*E*-03**	7.48*E*-03 ± 3.26*E*-02†	7.90*E*-02 ± 3.26*E*-02†
CILS	0.07	T_1	9.58*E*-03 ± 7.51*E*-02	**7.84*E*-04 ± 9.50*E*-05**†	8.06*E*-01 ± 2.35*E*-01†
		T_2	3.47*E*-04 ± 3.85*E*-05	**7.84*E*-04 ± 2.56*E*-06**	7.65*E*-04 ± 2.29*E*-06
PIHS	0.99	T_1	**7.12*E*-04 ± 2.74*E*-05**	4.16*E*-03 ± 3.24*E*-03†	7.35*E*-03 ± 4.26*E*-04†
		T_2	**5.17*E*-03 ± 6.30*E*-04**	5.37*E*-02 ± 7.71*E*-02†	5.17*E*-02 ± 8.92*E*-02†
PIMS	0.55	T_1	**3.28*E*-03 ± 3.52*E*-04**	5.19*E*-03 ± 7.42*E*-03	6.45*E*-03 ± 4.33*E*-03†
		T_2	**2.24*E*-02 ± 4.16*E*-03**	9.63*E*+01 ± 2.47*E*-00†	1.58*E*-01 ± 3.65*E*-00†
PILS	0.002	T_1	**1.85*E*-04 ± 1.10*E*-05**	5.32*E*-04 ± 1.15*E*-05	2.83*E*-04 ± 4.02*E*-04
		T_2	**3.32*E*-02 ± 1.07*E*-04**	8.54*E*-02 ± 2.33*E*-03	6.62*E*-01 ± 1.24*E*-04†
NIHS	0.94	T_1	**4.25*E*-02 ± 1.18*E*-02**	8.45*E*-00 ± 3.74*E*-02†	7.90*E*-01 ± 5.88*E*-01†
		T_2	**5.32*E*-05 ± 6.58*E*-06**	6.70*E*-04 ± 5.39*E*-04†	3.51*E*-04 ± 4.27*E*-04†
NIMS	0.51	T_1	**6.37*E*-02 ± 5.89*E*-03**	1.85*E*-01 ± 4.34*E*-01†	8.02*E*-01 ± 1.48*E*-01†
		T_2	**3.24*E*-03 ± 6.35*E*-04**	2.67*E*-02 ± 5.06*E*-02†	7.70*E*-02 ± 3.18*E*-02†
NILS	0.001	T_1	7.17*E*-03 ± 8.24*E*-04	**8.15*E*-04 ± 2.32*E*-05**†	9.06*E*-04 ± 3.11*E*-05†
		T_2	4.46*E*-00 ± 3.29*E*-01	**6.14*E*-01 ± 4.16*E*-04**†	6.54*E*-01 ± 4.23*E*-04†

sorting, it firstly develops job orders for the first procedure based on the list scheduling principle and then determines job sequences for the remaining procedures by a FIFO manner. For machine allocation, we use the rule of the first idle machine. The performance metric is based on the values of makespan, production cost, electricity consumption, water wastage, and processing cost, and the goal is to achieve the smallest values. The optimization results and manual scheduling results are summarized in Table 6.

It is seen from Table 6 that all optimization algorithms including the proposed MSO, MO-MFEA, and NSGA-II perform better than manual scheduling for all instances based on the smallest makespan value, the smallest production cost, the smallest electricity consumption value, the smallest water wastage value, and the smallest processing cost. The proposed MSO algorithm performs best for all of the instances except J4 and J8, for which MO-MFEA is the best because the approach itself is an excellent multitask evolutionary algorithm. Furthermore, out of 50 groups of data in Table 6, there are 49, 34, and 43 statistically significant differences between the proposed MSO and manual scheduling, the proposed MSO and MO-MFEA, and the proposed MSO and NSGA-II. Based on this result, the probability that the proposed MSO and compared algorithms are from the same distribution is low. The overall optimization results indicate that these optimization algorithms are high-efficiency ways, and the proposed MSO is the most promising approach for studied production scheduling with resource saving in textile printing and dyeing. The reason for the superior performance of the proposed MSO algorithm is that it increases interaction between subsystems and solution diversity by effectively using cross-subsystem migration.

To further investigate the potential of the proposed method in production scheduling with resource saving, saving ratio of each objective in each instance is shown in Table 7, which is got by, respectively, calculating the ratio of the values obtained by each optimization algorithm to the values obtained by manual scheduling. It is found from Table 7 that using optimization algorithms, the effect of production scheduling with resource saving including makespan, production cost, electricity consumption, water wastage, and processing cost is very significant for all instances. Furthermore, Figure 3 provides the visual presentation of mean saving ratio of all objectives of each instance. It is seen that for the proposed MSO algorithm, MO-MFEA, and NSGA-II, mean saving ratio is over 15%, 10%, and 6%, respectively, for the most of instances. Figure 4 provides the visual presentation of mean saving ratio of each objective for all instances. It is seen that for makespan, mean saving ratio is about 15%, 12%, and 6%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For production cost, mean saving ratio is about 19%, 14%, and 7%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For electricity consumption, mean saving ratio is about 15%, 13%, and 6%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For water wastage, mean saving ratio is about 17%, 15%, and 8%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For processing cost, mean saving ratio is about 17%, 15%, and 7%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. It indicates that for each performance metric for studied production scheduling with resource saving in textile printing and dyeing, the proposed MSO algorithm is better than other two compared algorithms. That is, the proposed MSO algorithm can not

Table 5: The number of jobs, procedures, and machines in each procedure, and the corresponding procedures of each instance in textile printing and dyeing.

Instance	Number of jobs	Number of procedures	Number of machines in each procedure	Procedures
J1	10	10	{2, 3, 4, 4, 3, 2, 3, 4, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–hot-melting–padding–stretching–shrinking–packaging
J2	10	7	{2, 4, 4, 3, 2, 3, 2}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J3	10	12	{2, 3, 3, 3, 2, 2, 3, 3, 2, 3, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J4	20	10	{2, 4, 4, 3, 3, 2, 3, 3, 2, 2}	Preparing–bleaching–setting–mercerizing–dyeing–drying–singeing–stretching–shrinking–packaging
J5	20	12	{3, 2, 2, 3, 4, 4, 2, 3, 2, 3, 2, 4}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J6	20	7	{2, 3, 4, 3, 2, 4, 2}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J7	30	10	{2, 3, 4, 2, 2, 4, 3, 3, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–hot-melting–padding–stretching–shrinking–packaging
J8	30	7	{2, 3, 3, 4, 4, 2, 4}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J9	30	12	{3, 2, 2, 4, 2, 4, 3, 4, 2, 3, 2, 2}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J10	30	8	{2, 4, 4, 3, 2, 3, 4, 3}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–shrinking–packaging

Table 6: Result comparisons between manual scheduling, the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

Instance	Objective functions	Manual scheduling	Optimization algorithms		
			MSO	MO-MFEA	NSGA-II
J1	Makespan	$1.45E+03 \pm 2.12E+01^{\dagger}$	$\mathbf{1.22E+03 \pm 5.90E+01}$	$1.31E+03 \pm 2.86E+01$	$1.36E+03 \pm 6.74E+01$
	Production cost	$1.49E+06 \pm 5.34E+04^{\dagger}$	$\mathbf{9.98E+05 \pm 2.14E+03}$	$1.05E+06 \pm 5.71E+04^{\dagger}$	$1.40E+06 \pm 7.58E+04^{\dagger}$
	Electricity consumption	$6.11E+04 \pm 8.22E+02^{\dagger}$	$\mathbf{5.01E+04 \pm 3.27E+02}$	$5.35E+04 \pm 3.05E+02^{\dagger}$	$5.78E+04 \pm 3.65E+02^{\dagger}$
	Water wastage	$9.84E+03 \pm 5.37E+02^{\dagger}$	$\mathbf{8.15E+03 \pm 2.36E+01}$	$8.61E+03 \pm 4.49E+01^{\dagger}$	$9.25E+03 \pm 1.18E+01^{\dagger}$
	Processing cost	$1.41E+06 \pm 5.36E+04^{\dagger}$	$\mathbf{1.12E+06 \pm 1.75E+04}$	$1.24E+06 \pm 2.35E+04$	$1.29E+06 \pm 2.36E+04$
J2	Makespan	$7.85E+02 \pm 6.34E+01^{\dagger}$	$\mathbf{6.91E+02 \pm 4.32E+01}$	$7.05E+02 \pm 1.18E+01^{\dagger}$	$7.10E+02 \pm 4.16E+01^{\dagger}$
	Production cost	$3.31E+05 \pm 7.48E+03^{\dagger}$	$\mathbf{2.75E+05 \pm 7.78E+03}$	$3.03E+05 \pm 5.21E+03^{\dagger}$	$3.12E+05 \pm 5.42E+03^{\dagger}$
	Electricity consumption	$2.74E+04 \pm 4.02E+02^{\dagger}$	$\mathbf{2.29E+04 \pm 6.63E+02}$	$2.48E+04 \pm 3.16E+02^{\dagger}$	$2.59E+04 \pm 3.60E+02^{\dagger}$
	Water wastage	$4.05E+03 \pm 2.30E+01^{\dagger}$	$\mathbf{3.36E+03 \pm 9.17E+01}$	$3.56E+03 \pm 7.06E+01^{\dagger}$	$3.80E+03 \pm 8.82E+01^{\dagger}$
	Processing cost	$3.05E+05 \pm 6.32E+03^{\dagger}$	$\mathbf{2.51E+05 \pm 2.64E+03}$	$2.79E+05 \pm 4.23E+03^{\dagger}$	$2.91E+05 \pm 1.16E+03^{\dagger}$
J3	Makespan	$7.08E+02 \pm 5.31E+01^{\dagger}$	$\mathbf{5.87E+02 \pm 3.62E+01}$	$6.04E+02 \pm 7.18E+01^{\dagger}$	$6.50E+02 \pm 5.42E+01^{\dagger}$
	Production cost	$6.31E+05 \pm 8.55E+03^{\dagger}$	$\mathbf{5.04E+05 \pm 1.31E+03}$	$5.51E+05 \pm 5.56E+03^{\dagger}$	$5.71E+05 \pm 3.18E+03^{\dagger}$
	Electricity consumption	$7.54E+04 \pm 4.60E+02^{\dagger}$	$\mathbf{6.28E+04 \pm 3.47E+02}$	$6.32E+04 \pm 3.17E+02$	$6.95E+04 \pm 4.29E+02^{\dagger}$
	Water wastage	$9.31E+03 \pm 8.66E+01^{\dagger}$	$\mathbf{7.93E+03 \pm 6.36E+01}$	$8.19E+03 \pm 2.36E+01^{\dagger}$	$8.67E+03 \pm 8.15E+01^{\dagger}$
	Processing cost	$5.25E+05 \pm 4.14E+03^{\dagger}$	$\mathbf{4.35E+05 \pm 7.89E+03}$	$4.61E+05 \pm 4.42E+03^{\dagger}$	$4.73E+05 \pm 6.63E+03^{\dagger}$
J4	Makespan	$1.76E+03 \pm 7.12E+01^{\dagger}$	$1.53E+03 \pm 4.35E+01$	$\mathbf{1.51E+03 \pm 1.15E+01}$	$1.60E+03 \pm 7.24E+01$
	Production cost	$8.12E+05 \pm 8.56E+03^{\dagger}$	$7.05E+05 \pm 6.69E+03$	$\mathbf{7.01E+05 \pm 3.27E+03}$	$7.35E+05 \pm 3.28E+03^{\dagger}$
	Electricity consumption	$4.51E+04 \pm 6.10E+02^{\dagger}$	$3.94E+04 \pm 3.78E+02$	$\mathbf{3.85E+04 \pm 8.13E+02}$	$4.00E+04 \pm 6.12E+02$
	Water wastage	$8.05E+03 \pm 5.23E+01^{\dagger}$	$7.00E+03 \pm 1.47E+01$	$\mathbf{6.90E+03 \pm 9.74E+01}$	$7.41E+03 \pm 6.69E+01^{\dagger}$
	Processing cost	$5.37E+05 \pm 4.16E+03^{\dagger}$	$4.89E+05 \pm 4.58E+03$	$\mathbf{4.78E+05 \pm 6.32E+03}$	$4.91E+05 \pm 4.42E+03^{\dagger}$
J5	Makespan	$1.72E+03 \pm 1.59E+01^{\dagger}$	$\mathbf{1.44E+03 \pm 6.61E+01}$	$1.56E+03 \pm 1.18E+01$	$1.58E+03 \pm 8.15E+01^{\dagger}$
	Production cost	$9.96E+05 \pm 8.44E+03^{\dagger}$	$\mathbf{8.21E+05 \pm 3.64E+03}$	$8.76E+05 \pm 2.35E+03^{\dagger}$	$9.25E+05 \pm 6.64E+03^{\dagger}$
	Electricity consumption	$7.84E+04 \pm 3.24E+02^{\dagger}$	$\mathbf{6.70E+04 \pm 8.96E+02}$	$6.92E+04 \pm 4.43E+02^{\dagger}$	$7.30E+04 \pm 2.01E+02^{\dagger}$
	Water wastage	$1.22E+04 \pm 1.10E+02^{\dagger}$	$\mathbf{9.81E+03 \pm 7.78E+01}$	$9.99E+03 \pm 1.06E+01^{\dagger}$	$1.13E+04 \pm 3.68E+02^{\dagger}$
	Processing cost	$8.94E+05 \pm 2.36E+03^{\dagger}$	$\mathbf{7.44E+05 \pm 5.93E+03}$	$7.63E+05 \pm 4.45E+03^{\dagger}$	$8.32E+05 \pm 1.17E+03^{\dagger}$
J6	Makespan	$1.07E+03 \pm 4.41E+01$	$\mathbf{1.01E+03 \pm 1.74E+01}$	$1.04E+03 \pm 7.78E+01$	$1.06E+03 \pm 5.24E+01$
	Production cost	$2.96E+05 \pm 8.39E+03^{\dagger}$	$\mathbf{2.49E+05 \pm 2.36E+03}$	$2.65E+05 \pm 5.36E+03^{\dagger}$	$2.75E+05 \pm 3.32E+03^{\dagger}$
	Electricity consumption	$2.62E+04 \pm 7.45E+02^{\dagger}$	$\mathbf{2.15E+04 \pm 4.47E+02}$	$2.32E+04 \pm 1.13E+02^{\dagger}$	$2.48E+04 \pm 4.40E+02^{\dagger}$
	Water wastage	$4.21E+03 \pm 8.36E+01^{\dagger}$	$\mathbf{3.29E+03 \pm 3.62E+01}$	$3.66E+03 \pm 7.58E+01^{\dagger}$	$4.01E+03 \pm 3.65E+01^{\dagger}$
	Processing cost	$1.56E+05 \pm 9.60E+03^{\dagger}$	$\mathbf{1.29E+05 \pm 1.18E+03}$	$1.42E+05 \pm 6.31E+03^{\dagger}$	$1.47E+05 \pm 2.47E+03^{\dagger}$
J7	Makespan	$1.71E+03 \pm 2.36E+01^{\dagger}$	$\mathbf{1.45E+03 \pm 6.35E+01}$	$1.51E+03 \pm 6.35E+01$	$1.62E+03 \pm 1.29E+01^{\dagger}$
	Production cost	$1.11E+06 \pm 6.03E+04^{\dagger}$	$\mathbf{9.31E+05 \pm 9.68E+03}$	$9.79E+05 \pm 4.47E+03^{\dagger}$	$1.02E+06 \pm 4.33E+04^{\dagger}$
	Electricity consumption	$3.67E+04 \pm 5.60E+02^{\dagger}$	$\mathbf{3.10E+04 \pm 7.41E+02}$	$3.23E+04 \pm 1.18E+02^{\dagger}$	$3.45E+04 \pm 9.54E+02^{\dagger}$
	Water wastage	$1.18E+04 \pm 8.15E+02^{\dagger}$	$\mathbf{9.73E+03 \pm 3.26E+01}$	$9.98E+03 \pm 5.69E+01^{\dagger}$	$1.03E+04 \pm 6.37E+02^{\dagger}$
	Processing cost	$1.16E+06 \pm 7.34E+04^{\dagger}$	$\mathbf{9.60E+05 \pm 5.59E+03}$	$9.98E+05 \pm 6.37E+03^{\dagger}$	$1.07E+06 \pm 2.50E+04^{\dagger}$
J8	Makespan	$1.53E+03 \pm 8.56E+01^{\dagger}$	$1.32E+03 \pm 5.23E+01$	$\mathbf{1.30E+03 \pm 2.28E+01}$	$1.42E+03 \pm 4.31E+01$
	Production cost	$4.16E+05 \pm 2.33E+03^{\dagger}$	$3.58E+05 \pm 1.82E+03$	$\mathbf{3.51E+05 \pm 1.25E+03}$	$3.82E+05 \pm 7.89E+03^{\dagger}$
	Electricity consumption	$5.53E+04 \pm 6.34E+02^{\dagger}$	$4.87E+04 \pm 4.34E+02$	$\mathbf{4.68E+04 \pm 4.13E+02}^{\dagger}$	$5.35E+04 \pm 1.12E+02^{\dagger}$
	Water wastage	$7.62E+03 \pm 7.79E+01^{\dagger}$	$6.71E+03 \pm 8.29E+01$	$\mathbf{6.45E+03 \pm 8.75E+01}^{\dagger}$	$7.02E+03 \pm 5.36E+01^{\dagger}$
	Processing cost	$3.91E+05 \pm 4.18E+03^{\dagger}$	$3.37E+05 \pm 1.44E+03$	$\mathbf{3.30E+05 \pm 5.36E+03}$	$3.63E+05 \pm 8.92E+03^{\dagger}$
J9	Makespan	$1.56E+03 \pm 6.40E+01^{\dagger}$	$\mathbf{1.33E+03 \pm 2.54E+01}$	$1.36E+03 \pm 1.23E+01$	$1.40E+03 \pm 1.23E+01$
	Production cost	$2.28E+06 \pm 8.23E+04^{\dagger}$	$\mathbf{1.78E+06 \pm 1.98E+04}$	$1.92E+06 \pm 7.55E+04^{\dagger}$	$2.02E+06 \pm 5.51E+04^{\dagger}$
	Electricity consumption	$4.92E+04 \pm 5.26E+02^{\dagger}$	$\mathbf{4.30E+04 \pm 6.65E+02}$	$4.37E+04 \pm 1.34E+02$	$4.46E+04 \pm 4.33E+02^{\dagger}$
	Water wastage	$1.98E+04 \pm 7.41E+02^{\dagger}$	$\mathbf{1.66E+04 \pm 1.27E+02}$	$1.77E+04 \pm 9.25E+02^{\dagger}$	$1.75E+04 \pm 7.75E+02^{\dagger}$
	Processing cost	$1.89E+06 \pm 2.35E+04^{\dagger}$	$\mathbf{1.52E+06 \pm 3.65E+04}$	$1.58E+06 \pm 3.11E+04$	$1.72E+06 \pm 8.24E+04^{\dagger}$
J10	Makespan	$2.21E+03 \pm 5.01E+01^{\dagger}$	$\mathbf{1.83E+03 \pm 4.42E+01}$	$2.03E+03 \pm 7.42E+01^{\dagger}$	$2.19E+03 \pm 5.61E+01^{\dagger}$
	Production cost	$3.30E+05 \pm 8.42E+03^{\dagger}$	$\mathbf{2.74E+05 \pm 1.12E+03}$	$3.02E+05 \pm 6.35E+03^{\dagger}$	$3.29E+05 \pm 8.76E+03^{\dagger}$
	Electricity consumption	$4.77E+04 \pm 7.41E+02^{\dagger}$	$\mathbf{3.95E+04 \pm 7.14E+02}$	$4.19E+04 \pm 8.17E+02^{\dagger}$	$4.67E+04 \pm 5.32E+02^{\dagger}$
	Water wastage	$9.21E+03 \pm 8.56E+01^{\dagger}$	$\mathbf{7.66E+03 \pm 8.23E+01}$	$8.11E+03 \pm 3.62E+01^{\dagger}$	$8.26E+03 \pm 9.01E+01^{\dagger}$
	Processing cost	$2.67E+05 \pm 9.62E+03^{\dagger}$	$\mathbf{2.23E+05 \pm 3.31E+03}$	$2.44E+05 \pm 4.28E+03^{\dagger}$	$2.66E+05 \pm 5.27E+03^{\dagger}$

only obtain better scheduling strategy but also reduce more electricity consumption and water wastage.

These results show that the proposed integrated optimization model and the corresponding solutions have a great potential in production scheduling with resource savings for textile printing and dyeing industry. Many manufacturing industries, such as papermaking manufacturing and glass manufacturing, are both energy-intensive and material-intensive. Production in these manufacturing often wastes a lot of energy and material, so

Table 7: Saving ratio comparisons between manual scheduling, the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

Instance	Objective functions	Saving ratio (%)		
		MSO	MO-MFEA	NSGA-II
J1	Makespan	15.86	9.66	6.21
	Production cost	33.02	29.53	6.04
	Electricity consumption	18.00	12.44	5.40
	Water wastage	17.17	12.50	6.00
	Processing cost	20.57	12.06	8.51
J2	Makespan	11.97	11.08	9.55
	Production cost	16.92	8.46	5.74
	Electricity consumption	16.42	12.04	5.47
	Water wastage	17.04	12.10	6.17
	Processing cost	17.70	11.80	4.59
J3	Makespan	17.09	14.69	8.19
	Production cost	20.13	12.68	9.51
	Electricity consumption	16.71	16.18	7.82
	Water wastage	14.82	12.03	6.87
	Processing cost	17.14	12.19	9.90
J4	Makespan	13.07	14.20	9.09
	Production cost	13.18	13.67	9.48
	Electricity consumption	12.64	14.63	11.31
	Water wastage	13.04	14.29	7.95
	Processing cost	8.94	10.99	8.58
J5	Makespan	16.28	11.63	8.14
	Production cost	17.57	12.05	7.13
	Electricity consumption	14.54	11.73	6.89
	Water wastage	19.59	19.36	7.38
	Processing cost	16.78	14.65	6.94
J6	Makespan	5.61	2.80	0.93
	Production cost	15.88	10.47	7.09
	Electricity consumption	17.94	11.45	5.34
	Water wastage	21.85	13.06	4.75
	Processing cost	17.31	15.38	5.77
J7	Makespan	15.20	11.70	5.26
	Production cost	16.13	11.80	8.11
	Electricity consumption	15.53	11.99	5.99
	Water wastage	17.54	15.42	12.71
	Processing cost	17.24	13.97	7.76
J8	Makespan	13.73	15.03	7.19
	Production cost	13.94	15.63	8.17
	Electricity consumption	13.74	15.37	3.25
	Water wastage	14.57	15.35	7.87
	Processing cost	13.81	15.60	7.16
J9	Makespan	14.74	12.82	10.26
	Production cost	21.93	15.79	11.40
	Electricity consumption	12.60	11.18	9.35
	Water wastage	16.16	10.61	11.62
	Processing cost	19.58	16.40	8.99
J10	Makespan	17.19	12.67	0.90
	Production cost	16.67	8.48	0.30
	Electricity consumption	17.19	12.16	2.10
	Water wastage	16.83	11.94	10.31
	Processing cost	16.48	12.36	0.37

the decision makers tend to reduce them during the production process. This study can serve as a reference for these manufacturers who are interested in improving energy and material savings through production scheduling, and the proposed MSO algorithm can be an attractive method for solving these complex multisystem problems.

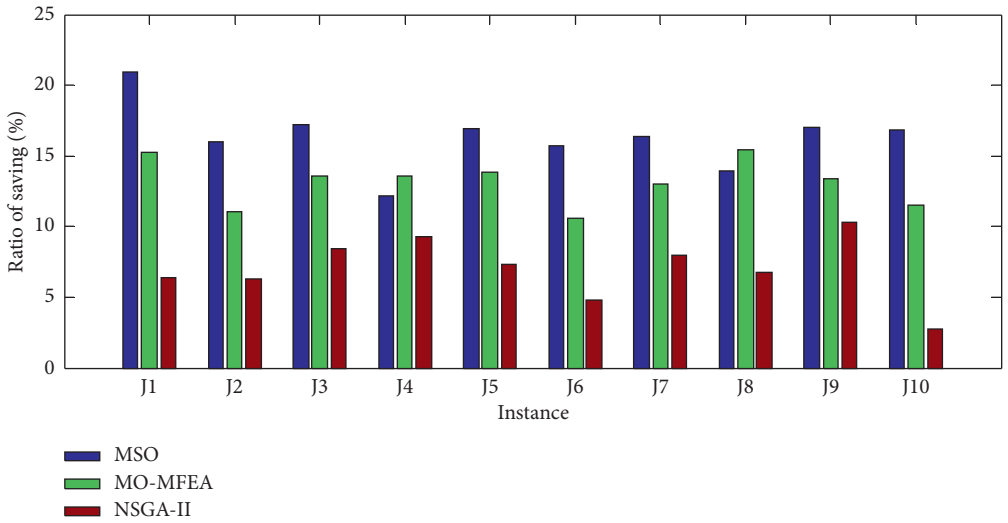


Figure 3: The mean saving ratio of all objectives of each instance obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

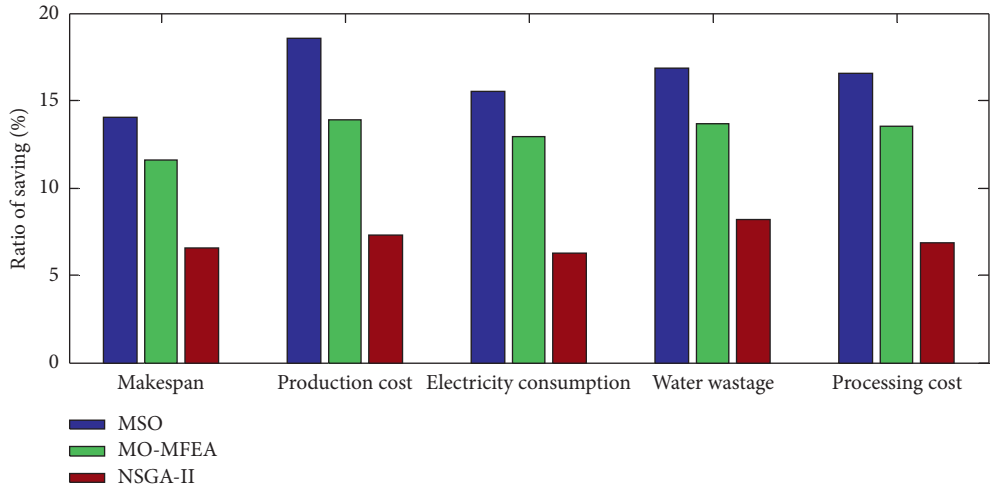


Figure 4: The mean saving ratio of each objective for all instances obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

6. Conclusions

In this paper, an integrated optimization model for textile printing and dyeing is first built by considering both production efficiency and resource saving, which is formulated as a multisystem optimization problem. It includes a production scheduling subsystem and a resource saving subsystem, and each of the subsystems contains multiple objectives. Then, a multisystem optimization algorithm called the MSO algorithm, composed of within-subsystem evolution and cross-subsystem migration operators, is proposed to solve general multisystem optimization problem. The performance of the MSO algorithm is investigated on a set of benchmark functions, and the numerical simulations show that the proposed MSO algorithm is better than MO-MFEA and NSGA-II for the most of the benchmark functions.

Finally, the MSO algorithm is applied to the proposed integrated production scheduling with resource saving problem, and the results again show that the proposed MSO algorithm is a competitive multisystem optimization method. In particular, the MSO algorithm is better than its constituent NSGA-II on all the test instances, and it is better than MO-MFEA on 8 of 10 instances.

The proposed MSO algorithm is a heuristic method for complex multisystem optimization. The complex multisystem contains multiple subsystems, and each of the subsystems may contain multiple objectives and multiple constraints. Compared to single-objective and multiobjective optimization algorithms, the proposed MSO algorithm has a more complex structure that is well-suited for real-world problems. Based on the optimization tests in this paper, the MSO algorithm can effectively solve multisystem optimization problems, including real-world textile printing and dyeing scheduling with resource saving. The MSO framework presented here could be extended for other types of optimization algorithms. Moreover, real-world textile

printing and dyeing scheduling problem with resource saving is formulated in a way that is amenable to the multisystem optimization algorithm, which serves as a reference for other manufacturing with similar production processes to promote in resource saving.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Zhejiang Provincial Natural Science Foundation of China under grant no. LY19F030011, National Natural Science Foundation of China under grant nos. 52077213 and 62003332, and Natural Science Foundation of Guangdong Province under grant no. 2018A030310671. H. Zhou was supported by the UK EPSRC under grant no. EP/N011074/1, Royal Society-Newton Advanced Fellowship under grant no. NA160342, and European Union's Horizon 2020 Research and Innovation Program under the Marie-Sklodowska-Curie grant agreement no. 720325.

References

[1] Y. Lu, T. Peng, and X. Xu, "Energy-efficient cyber-physical production network: Architecture and technologies," *Computers & Industrial Engineering*, vol. 129, pp. 56–66, 2019.

[2] M. Vieira, T. Pinto-Varela, and A. P. Barbosa-Póvoa, "A model-based decision support framework for the optimisation of production planning in the biopharmaceutical industry," *Computers & Industrial Engineering*, vol. 129, pp. 354–367, 2019.

[3] S. Ye, H. Ma, S. Xu et al., "An effective fireworks algorithm for warehouse-scheduling problem," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 1, pp. 75–85, 2017.

[4] F. Li, L. Zhou, G. Xu et al., "An empirical study on solving an integrated production and distribution problem with a hybrid strategy," *PloS One*, vol. 13, no. 11, Article ID e0206806, 2017.

[5] H. Ma, S. Su, D. Simon, and M. Fei, "Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling," *Engineering Applications of Artificial Intelligence*, vol. 44, pp. 79–90, 2015.

[6] Z. Zeng, M. Hong, Y. Man, J. Li, Y. Zhang, and H. Liu, "Multi-object optimization of flexible flow shop scheduling with batch process - consideration total electricity consumption and material wastage," *Journal of Cleaner Production*, vol. 183, pp. 925–939, 2018.

[7] M. Biondi, G. Sand, and I. Harjunkoski, "Optimization of multipurpose process plant operations: a multi-time-scale maintenance and production scheduling approach," *Computers & Chemical Engineering*, vol. 99, pp. 325–339, 2017.

[8] L. S. Dias, R. C. Pattison, C. Tsay, M. Baldea, and M. G. Ierapetritou, "A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units," *Computers & Chemical Engineering*, vol. 113, pp. 139–151, 2018.

[9] J. Allison, *Complex System Optimization: A Review of Analytical Target Cascading, Collaborative Optimization, and Other Formulations*, M. S. Thesis, University of Michigan, Ann Arbor, MI, USA, 2004.

[10] J. Abell and D. Du, "A framework for multi-objective, biogeography-based optimization of complex system families," in *Proceedings of the AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, pp. 1–8, Fort Worth, TX, USA, 2010.

[11] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[12] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[13] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[14] L. M. Antonio and C. A. Coello, "Coevolutionary multi-objective evolutionary algorithms: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 851–865, 2018.

[15] A. Gupta, Y. S. Ong, L. Feng et al., "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[16] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: a computer science view of cognitive multitasking," *Cognitive Computation*, vol. 8, no. 2, pp. 125–142, 2016.

[17] D. Du and D. Simon, "Complex system optimization using biogeography-based optimization," *Mathematical Problems in Engineering*, vol. 8, Article ID 456232, , 2013.

[18] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[19] H. Ma, M. Fei, Z. Jiang, L. Li, H. Zhou, and D. Crookes, "A multipopulation-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 689–702, 2020.

[20] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 590–605, 2016.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[22] R. J. Hathaway and J. C. Bezdek, "Fuzzy c-means clustering of incomplete data," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 5, pp. 735–744, 2001.

[23] Y. Yuan, Y. S. Ong, L. Feng et al., "Evolutionary multitasking for multiobjective continuous optimization: benchmark problems, performance metrics and baseline results," Technical Report, pp. 1–10, Cornell University, Ithaca, NY, USA, 2016.

[24] Z. Zhao, B. Liu, C. Zhang, and H. Liu, "An improved adaptive NSGA-II with multi-population algorithm," *Applied Intelligence*, vol. 49, no. 2, pp. 569–580, 2019.

Hindawi
Complexity
Volume 2020, Article ID 8877008, 11 pages
https://doi.org/10.1155/2020/8877008

Research Article

Knee Point-Guided Multiobjective Optimization Algorithm for Microgrid Dynamic Energy Management

Wenhua Li,[1] **Guo Zhang**,[1] **Tao Zhang**,[1,2] **and Shengjun Huang**[1,2]

[1]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*Hunan Key Laboratory of Multi-Energy System Intelligent Interconnection Technology, Changsha 410073, China*

Correspondence should be addressed to Shengjun Huang; huangshengjun@nudt.edu.cn

Received 25 September 2020; Revised 16 October 2020; Accepted 21 October 2020; Published 7 November 2020

Academic Editor: Shi Cheng

Model predictive control (MPC) technology can effectively reduce the bad effect caused by inaccurate data prediction in microgrid energy management problem. However, the use of MPC technology needs to dynamically select an optimal solution from the Pareto solution set to implement, which needs the participant of the decision-makers frequently. In order to reduce the burden on decision-makers, we designed a knee point-based evolutionary multiobjective optimization algorithm, termed KBEMO. Knee point is the solution on Pareto front with the maximum marginal utility, which is considered as the preferred solution if there is no other preference. This algorithm focuses on obtaining the knee region and automatically outputs knee points after the optimization. By combining this algorithm with MPC technology, it can effectively reduce the amount of computational consumption and obtain better convergence. Experimental results show that this method is more competitive than the traditional single-objective MPC method.

1. Introduction

In recent years, energy shortages and environmental pollution problems have become increasingly severe. In order to improve the safety and survivability of traditional power grid, microgrid technology has received more and more attention, especially in absorbing large-scale wind and solar energy into the power grid, which has great application prospects. The concept of microgrid was first proposed by the American Electric Power Technology Research Institute, the Consortium for Electric Reliability Technology Solutions (CERT) [1]. It is a small power supply and distribution autonomous system that contains a variety of microgrid sources, e.g., electrical loads, energy storage devices, and energy conversion devices. The microgrid can be self-regulated, monitored, and self-protected [2, 3].

The uncertainty of renewable energy in the microgrid increases the risk of system scheduling. How to adapt to the uncertain working conditions of various renewable energy output and realize the benign interaction of multiple resources and economic operation is a difficult task of the research in energy management and optimization [4]. The main problem in energy management is how to formulate a power generation plan and dispatch plan with the lowest overall operating cost under the stable operation of the system in response to the uncertainty in the process of power generation and power consumption so as to meet the needs of production and life better. With the development of microgrid technology, more and more energy supply and energy-consuming equipment are connected to the microgrid. However, renewable energy power generations such as wind and solar power and users' energy demand are random and uncertain. Existing studies have shown that the method of day-ahead scheduling [5] is useful. Specifically, energy management optimization is performed based on the forecast results of the next 24 hours. However, the result often differs from the actual situation. In response to this problem, Zhang et al. [6] introduced a model predictive control (MPC) framework to optimize the microgrid energy management problem. The main process can be summarized as follows: (1) in each period, the microgrid system will predict the renewable energy power generation and user

load of the next 24 hours; (2) optimize the operation based on the predicted data; (3) apply the optimal solution obtained at the current moment and run the system according to real-world data; (4) update the status of the microgrid system; and (5) return to the first step and repeat the whole process. The MPC framework [7] continuously optimizes the energy management problem during the continuous advancement of the period, thereby reducing the impact of uncertain factors. This method has been successfully applied to many microgrid energy management problems, and a high-quality solution has been achieved.

Microgrid energy management is a typical multiobjective optimization problem (MOP). We need to consider optimizing more than one objective when dealing with the microgrid energy management problem, e.g., the loss of energy storage system, stability of power supply, energy-saving, and pollutant emission reduction [8]. To deal with MOPs, the traditional linear programming method generally adopts the method of weighted sum (WS); that is, the decision-maker gives a set of weight vectors in advance and then couples the objective vector and the weight vector into a scalar function for optimization. Although this method is simple to apply, it has many problems. On the one hand, due to the inconsistency of the scales between different objective functions, it is difficult to determine the exact value of the weight vector, resulting in poor robustness when optimizing the single-objective optimization problem. Specifically, the use of larger weights will increase the influence of the objective function noise, leading to unilateral preference. On the other hand, it is often difficult for decision-makers to give his appropriate preference weights in advance.

To address the issue of using the WS method to solve MOP, evolutionary multiobjective optimization (EMO) has attracted more and more attention [9, 10]. The EMO algorithm can give a satisfactory solution set when the number of objective functions is 2 or 3 and has been successfully applied to solve a large number of real-world engineering problems [11]. Without loss of generality, the definition of a multiobjective optimization problem is as follows:

$$\begin{aligned} \text{Minimize, } & F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\}, \\ \text{s.t., } & \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega, \end{aligned} \tag{1}$$

where Ω denotes the feasible decision space, m is the number of objectives, and $\mathbf{x}$ is a decision vector consisting of n decision variables x_i. A solution $\mathbf{x_a}$ is said to Pareto dominate another solution $\mathbf{x_b}$ if and only if $\forall i = 1, 2, \ldots, m$, $f_i(\mathbf{x_a}) \leq f_i(\mathbf{x_b})$ and $\exists j = 1, 2, \ldots, m$, $f_j(\mathbf{x_a}) < f_j(\mathbf{x_b})$. The images of all Pareto optimal solution sets (PSs) in the objective space are termed the Pareto optimal front (PF).

Generally speaking, the EMO algorithm will give a set of Pareto optimal solution that does not dominate each other and no solution can outperform other solutions on all objective functions. Since it is always necessary to select a solution from many Pareto optimal solutions to apply in real-world engineering problems, general multiobjective optimization problems need to be combined with corresponding decision-making methods. When the number of goals is small, such as 2-3 goals, the decision-makers can intuitively select the appropriate solution from the Pareto front according to their own preferences by plotting all solutions. However, it is pointed out from cognitive studies that when the number of objectives is greater than 5, the applicability of this type of method is greatly reduced. In response to this problem, researchers have proposed many posterior methods, such as grey correlation analysis [12] and TOPSIS method [13].

Microgrid energy management combined with the MPC framework is a dynamic optimization problem. To solve this problem by the EMO algorithms, it is necessary to use a posteriori method to select an optimal solution for application after optimization, which will lead to the waste of computational resource. Generally speaking, the decision-maker is only interested in a small area of Pareto front, and other solutions on the Pareto front will waste a lot of computing resources. Therefore, focusing on certain representative areas of the Pareto front is of great significance for solving real-world engineering problems. Research points out that knee, as the point with the greatest marginal utility on the Pareto front, is more attractive to decision-makers [14]. For the two-objective optimization problem, the knee in the Pareto front refers to the solution with the largest marginal rate of return; that is, a small improvement in one objective will lead to serious degradation of at least another objective [15], as shown in Figure 1. Intuitively, the knee area is a "bulge" on the front of Pareto, where the slope changes suddenly. Many works have discussed the importance of knee and pointed out that, for general MOPs, if the decision-maker has no other preference, the knee can be used as the preferred solution.

Based on the above considerations, to solve the online microgrid energy management problem by the EMO algorithm, we first proposed a microgrid energy management mathematical model combined with the MPC framework. Moreover, a knee-based EMO algorithm (KBEMO) is proposed to solve the energy management problem, which can focus on obtaining the knee region on the Pareto front and give the global knee point as the final solution. Combined with this algorithm, the optimal solution is adopted continuously after each optimization process. Therefore, the microgrid can run as time goes, and the impact of renewable energy and user load uncertainty in the microgrid can be reduced. As a result, the running cost of the microgrid can be improved. Experiments show that this method can effectively obtain the optimal result of the microgrid energy management problem and reduce the impact of randomness.

The rest of this paper is organized as follows: Section 2 introduces the related models of microgrid energy management; Section 3 describes the knee-based EMO algorithm in detail and its application under the MPC framework; the experimental settings and experimental results are mentioned in Section 4; finally, we summarize the advantages and disadvantages of the current method and propose possible future research directions.

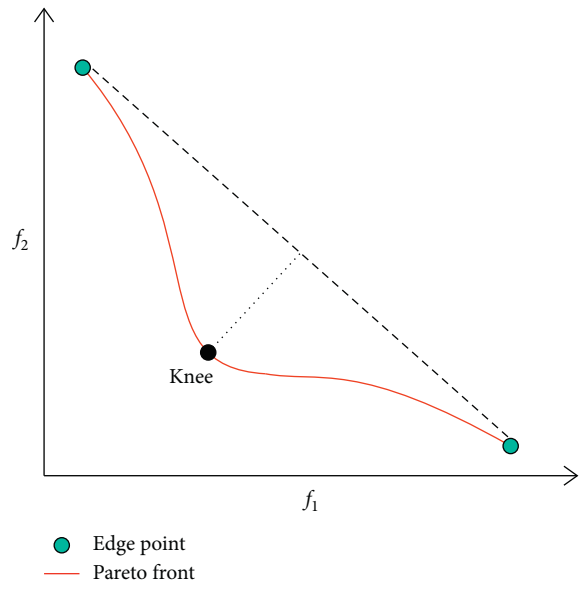


Figure 1: Illustration of knee point.

2. Microgrid Energy Management Model

Since this paper mainly discusses the application of the EMO algorithm in dynamic microgrid energy management, we select a typical microgrid model, which specifically includes renewable energy (wind and solar), energy storage system, user load, and the power grid. The time-of-use electricity price strategy is adopted, which means that the electricity price changes over time. Therefore, by introducing an energy storage system, we can purchase electricity from the power grid when the electricity price is low and sell electricity to the power grid when the price is high. Therefore, the overall operating cost of the microgrid can be reduced. Since there are many parameters involved in this model, the relevant parameters are summarized as follows (Table 1).

With the development of microgrid technology, more and more energy supply and energy-consuming equipment are introduced, and the model of microgrid energy management has become more and more complex. This article mainly studies the application of knee-based algorithms in dynamic energy management, so it considers the basic microgrid composition, including renewable energy power generation, energy storage systems, user loads, and large power grids.

As one of the most flexible and directly controllable devices in the microgrid, the energy storage system can not only be used as a consumer device to absorb excess electric energy but also as a backup power supply when the power generation is insufficient. Its constraints mainly include the maximum and minimum capacity limits and charge and discharge energy transfer power, which can be expressed as

$$P_{\text{bess}}^{\min} \le P_{\text{bess}}(k) \le P_{\text{bess}}^{\max}, \tag{2}$$

$$E_{\text{bess}}^{\min} \le E_{\text{bess}}(k) \le E_{\text{bess}}^{\max}, \tag{3}$$

$$E_{\text{bess}}(k+1) = E_{\text{bess}}(k) + \eta_{\text{bess}} P_{\text{bess}}(k)\Delta t. \tag{4}$$

Equations (2)–(4), respectively, represent the charge and discharge power constraints, energy level constraints, and dynamic energy transfer constraints of the energy storage system. Energy storage systems have different battery losses under the different state of charge (SOC), which can be expressed as

$$f_{\text{soc}}(k) = 1 - \frac{1}{1 + \alpha^* \exp\left(\beta^* \text{soc}(k)\right)}. \tag{5}$$

Among them, $\text{soc}(k) = E(k)/E_{\text{bess}}^{\max}$ represents the current state of the energy storage system and α and β characterize the internal parameters of the energy storage system, which is provided by the manufacturer. Similarly, the interaction between the microgrid system and the power grid also needs to meet the upper and lower power limits:

$$P_{\text{grid}}^{\min} \le P_{\text{grid}}(k) \le P_{\text{grid}}^{\max}. \tag{6}$$

The prerequisite for the economic dispatch of the microgrid is that the power supply and consumption of the microgrid system are always balanced, which is expressed as

$$P_{\text{solar}}(k) + P_{\text{grid}}(k) = P_{\text{bess}}(k) + P_{\text{load}}(k). \tag{7}$$

Many objective functions can be considered for microgrid energy management problems. We take the economic operation of the microgrid and the loss of the energy storage system as the optimization objectives, which can be expressed as

$$\begin{aligned} \min f_1 &= \sum C_{\text{grid}}(k) = \sum c_{\text{grid}}(k) P_{\text{grid}}(k)\Delta t, \\ \min f_2 &= \sum C_{\text{bess}}(k) = \sum f_{\text{soc}}(k). \end{aligned} \tag{8}$$

Among them, f_1 is the cost incurred by the interaction between the microgrid and the power grid and f_2 represents the loss of the energy storage system incurred during the operation.

The day-ahead scheduling technology of the microgrid means that, for the period that needs to be scheduled, the forecasting method is used to obtain data such as future renewable energy generation and user load. According to the data, the microgrid energy management is optimized, and the operation plan in the future period is obtained. In actual operation, it is difficult to accurately predict the data, so it will be adjusted according to the real-time data and the principle of power balance to achieve the purpose of stable operation.

3. Knee-Based Multiobjective Optimization Algorithm

3.1. Motivation and Framework. Generally speaking, the process of the algorithm based on the knee can be summarized as detecting the knee on the Pareto front and guiding the evolutionary search direction according to the knee information, so as to obtain the knee area. In recent years, researchers have proposed many methods on how to

Table 1: Parameter name and meaning.

Parameter type	Symbol	Meaning
System parameters	T	Control period of the model
	$P_{\text{grid}}^{\min}$ $P_{\text{grid}}^{\max}$	Upper and lower power limits of interaction with the power grid
	$E_{\text{bess}}^{\text{cur}}$	Initial energy storage level of the energy storage system
	$E_{\text{bess}}^{\max}$ $E_{\text{bess}}^{\min}$	Upper and lower limits of energy storage system capacity
	$P_{\text{bess}}^{\min}$ $P_{\text{bess}}^{\max}$	Upper and lower limits of charging power for energy storage system
	$\varepsilon_{\text{bess}}$	Self-discharge energy loss of energy storage system
	η_{bess}	Energy storage system charging and discharging efficiency
Predictive variable	$P_{\text{solar}}(k)$	Power of PV prediction generation
	$P_{\text{load}}(k)$	Power of load prediction generation
Decision variable	$\delta_{\text{bess}}^{c}(k)$	Energy storage system charge and discharge status
	$P_{\text{bess}}^{c}(k)$	Energy storage system charge and discharge power
Cost	$C_{\text{bess}}(k)$	Energy storage system operation and maintenance costs
	$C_{\text{grid}}(k)$	Interaction costs with large grids

detect knee. Das [16] proposed a knee point detection method based on the normal boundary intersection (NBI). Das defines an extreme line for the two-objective problem, called individual minimal convex hull (CHIM), which passes through the two boundary points (also called edge points) of the Pareto front. As shown in Figure 1, the knee point corresponds to the solution furthest from CHIM. This method is also known as a distance-based knee detection method and has been successfully applied to algorithms such as KnEA [14] and k-NSGA-II [9].

The existing knee detection methods can be mainly divided into two categories: (1) detection based on the geometric characteristics of the Pareto front. Unlike other parts of the Pareto front, the knee area has obvious geometric characteristics, that is, a curvature of the Pareto front. Representative methods include distance-based method [16], minimum Manhattan distance [17] (MMD), and reflection angle [15]. (2) Use Pareto front to evaluate the trade-offs between solutions. In the knee region, the influence between the objective function values increases. The knee region is detected by calculating such influences. Representative methods include cone-domination [18], trade-off [19], and expected marginal utility [15] (EMU) method. The performance of each method in knee detection is compared in [20] in detail, which pointed out that the distance-based detection method has strong robustness. However, the distance-based method is highly sensitive to the position of the boundary point, and different algorithms have different determinations of the Pareto front boundary. Therefore, this method may cause the problem of inaccurate knee detection.

The minimum Manhattan distance (MMD) method calculates the Manhattan distance of each solution on the Pareto front, and the solution with the minimum Manhattan distance is the current global knee point. The calculation method of this method is simple and has strong robustness. Therefore, we adopt the MMD method to detect the knee. The specific steps are as follows:

(1) Standardize the objective function values of the currently obtained solutions $x^{(1)}, \ldots, x^{(k)}$:

$$f'_m\left(x^{(i)}\right) = \frac{f_m\left(x^{(i)}\right) - z_m^*}{z_m^{\text{nad}} - z_m^*}, \tag{9}$$

where z^* and z^{nad} are the best and worst values of all Pareto optimal solutions and f_m and f'_m are the original target value and the normalized target value of the m-th dimension, respectively.

(2) Calculate the Manhattan distance of the solution $x^{(i)}$ on the Pareto front:

$$f'_1\left(x^{(i)}\right) + f'_2\left(x^{(i)}\right) + \cdots + f'_M\left(x^{(i)}\right) = v. \tag{10}$$

(3) The solution with the smallest Manhattan distance is the knee point of the current evolutionary process.

It can be seen from Figure 2 that the Manhattan distance of the solution $x^{(i)}$ can be expressed as a straight line with a slope of -1, which passes through the solution $x^{(i)}$ and the intercept is v. This paper detects the knee based on the MMD method. After obtaining the current knee information, the information is used to guide the evolution direction of the population, and the final knee region is obtained based on the NSGA-II [21] algorithm. The algorithm framework is illustrated in Algorithm 1.

After the algorithm detects the knee, it calculates the environmental fitness value of all individuals based on the knee information (line 8). The calculation process can be expressed as follows:

(1) Calculate the crowding distance of all individuals $x^{(i)}$ according to the crowding distance $\text{cd}^{(i)}$ calculation method of the NSGA-II algorithm.

(2) If the Euclidean distance $D_{i,j}$ between the individual $x^{(i)}$ and the knee point in the objective space is less than the given threshold δ, the crowding distance of solution $x^{(i)}$ will be rewarded according to the following equation:

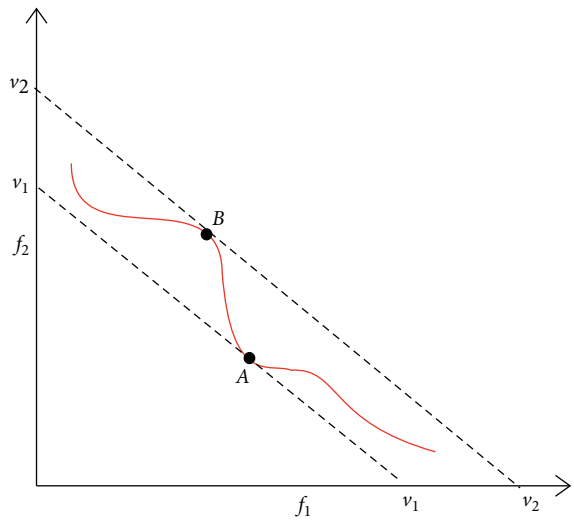


Figure 2: Illustration of the MMD method.

$$\mathrm{cd}^{(i)} = \mathrm{cd}^{(i)} \cdot \left(1 + \frac{1}{1 + \exp\left(D_{i,j} - \delta\right)}\right). \tag{11}$$

Among them, $1/1 + \exp(D_{i,j} - \delta)$ is the activation function; that is, when $D_{i,j} < \delta$, $\mathrm{cd}^{(i)}$ will be rewarded. The threshold δ is set by the user and is used to adjust the size of the knee area finally obtained.

Since the NSGA-II algorithm is likely to choose solutions with a larger crowding distance to form the next generation during the evolution, by increasing the crowding distance of individuals that are near to the current knee, the fitness value of such individuals is improved. In the process of evolution, the solutions around the knee are retained. On the other hand, by detecting the knee point during the evolution process, we can utilize the knee as the current elite individual. Moreover, the excellent genes of current knee solution will be shared with other solutions through crossover (line 5) to improve the convergence performance of the algorithm. It is worth noting that the edge point on the Pareto front of the multiobjective optimization problem plays a guiding role in obtaining the Pareto front. Besides, in some special situations, the decision-maker is interested in the solution of the boundary point, so the proposed algorithm treats the boundary point as knee points as well so that the boundary points can be retained during the evolution process.

3.2. Dynamic Optimization of Microgrid Energy Management. The day-ahead scheduling method predicted data for some time in the future (usually a day) to obtain the running strategy in advance. In actual running time, it will dynamically adjust based on actual data to satisfy the energy balance constraints. Although this method can make a reasonable control strategy for system operation to a certain extent, it has many disadvantages. Because it is difficult to accurately predict the renewable energy generation and user loads, in the actual operation process, the operating state of the system is also different from that of the preplanning stage. Applying solutions obtained based on inaccurate forecast data will make the actual running cost more expensive than the plan.

Model predictive control [19] will plan the operation strategy for a period of time in the future based on the predicting data and execute the obtained operation strategy at the current moment. At the next moment, the system state and the prediction for the future will be updated based on actual operating data, and the new planning operation strategy is applied at this moment. By continuously repeating the above process, the microgrid energy management system continues to obtain the operating strategy of the microgrid. Because this method continuously adjusts the optimizer based on actual data, it can reduce the impact of randomness and reduce the requirements for prediction accuracy.

After the EMO algorithm obtains the Pareto solution set, it needs to introduce the preference of the decision-maker to choose a solution for implementation. However, in dynamic optimization problems, it is unrealistic to frequently require decision-makers to participate, so we must find a way to automatically select a solution that satisfies the preference of decision-maker. The knee point on the Pareto front is the most "cost-effective" point in the entire solution set, which is regarded as the better solution when there is no other preference. In the model predictive control framework, the current global knee is selected as the optimal solution for implementation so that decision-makers do not need to provide preference information. It is worth noting that when there are multiple knees in the Pareto front of the problem, the algorithm will select the global knee as the optimal solution. The mainframe of the proposed method is shown in Figure 3.

It is worth noting that the optimization strategy of the current period has guiding significance for the next stage of optimization. In particular, when the forecast data are 100% accurate, the microgrid energy management strategy derived at the current moment should be also suitable for the next stage. In order to make better use of the optimization information obtained in the previous period and reduce the computational waste, we select solutions from the final population in the current stage to form up the elite individuals, which is combined with the initial population for the next stage. By doing this, the optimizer can converge to the Pareto optimal front faster. Specifically, the boundary points and the points around the knee point are chosen. The reason to select only a small part of the population is that it is easy to fall into the local optimal solution if we directly use the current population as the initial population. For the excellent individuals in the previous stage, the coded sequence 2 to T needs to be moved to the sequence 1 to $T-1$, and the value is randomly generated to fill the code T.

3.3. Experimental Parameter Setting. This paper proposed a microgrid energy management model including energy storage system, renewable energy power generation, user load, and power grid and selects a microgrid project as a case to verify the optimization method proposed in the

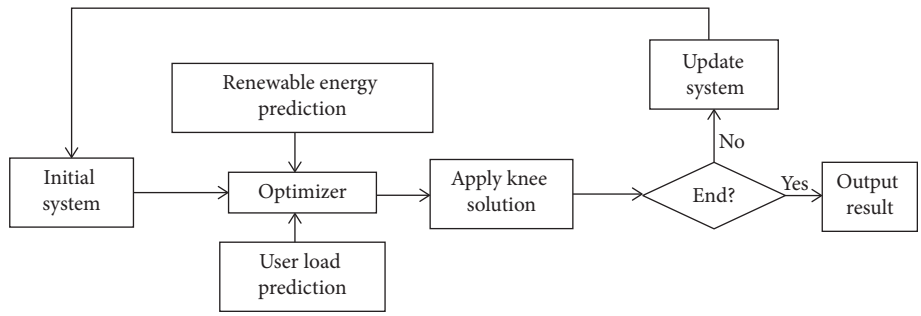


Figure 3: MPC framework optimization flow chart.

article. The main parameters of this microgrid are shown in Table 2. The relevant parameters of the knee-based EMO algorithm (hereinafter referred to as KBEMO algorithm) are shown in Table 3. For renewable energy power generation and user load data, we use the global energy forecasting competition 2014 data set and get the time-of-use electricity price based on the local electricity price, as shown in Figure 4.

4. Result and Analysis

4.1. Performance of the Proposed Algorithm. In order to exam the performance of the proposed KBEMO algorithm in searching and obtaining the knee area, we selected the classic knee test problems to test the KBEMO algorithm. Specifically, Branke et al. [15] proposed DEB2DK, DEB3DK, and DO2DK problems based on the DTLZ test problem [22], and the parameter K is introduced to control the number of knees on the Pareto front. We independently run 31 times for each test problem, calculate the generation distance (GD) of the results obtained in each run, and select the result closest to the average GD for display.

Figure 5 shows the Pareto fronts obtained by the KBEMO algorithm on DEB2DK, DEB3DK, and DO2DK problems. It can be seen from the figure that KBEMO can accurately detect and obtain the knee area on the Pareto front, and the boundary points of the Pareto front can be completely preserved. In terms of algorithm convergence, KBEMO can be very close to the true Pareto front. Therefore, the KBEMO algorithm can completely find and retain the knee area on the selected test problem, and it has great convergence performance.

4.2. Algorithm Performance Comparison. In order to compare the difference between the KBEMO algorithm and the classic multiobjective optimization algorithm, we compare the Pareto fronts obtained by KBEMO and MOEA/D-DE for the same problem. We choose MOEA/D-DE [23] to represent the classic multiobjective optimization algorithm, which has been proven to have strong competitiveness in dealing with multiobjective optimization problems. To ensure fairness, the two algorithms use the same operating parameters and run 31 times independently and compare the results that best represents their average level. The results are shown in Figure 6.

It can be seen from Figure 6 that the Pareto front obtained by the KBEMO algorithm is closer to the theoretical optimal value than the classic optimization algorithm MOEA/D-DE, which shows that the KBEMO algorithm can obtain a better convergence solution under the same conditions. At the same time, the algorithm accurately gives the global knee point for this problem. The solution represented by this point can well balance the two objective functions, which can better represent the preference of the decision-maker. For the traditional EMO algorithm, generally speaking, a larger population is required to describe the complete Pareto front, which greatly increases the waste of irrelevant calculations because the decision-maker does not need this information. Because the KBEMO algorithm only focuses on a specific area on the Pareto front, the required population is smaller and more advantageous, and because it focuses more on the search in a specific area, the algorithm has a stronger convergence ability than the normal algorithm.

4.3. Comparison of MPC Framework and Day-Ahead Scheduling. The method based on day-ahead scheduling (hereinafter referred to as DA) only needs to use the optimizer for a single-time optimization, so it has a great advantage in computational efficiency. However, the scheduling method obtained by this method is based on the premise of an accurate prediction of future data and the actual situation is also different from the forecast data. To compare the difference between the day-ahead scheduling and the MPC-based framework, we conduct energy management for the microgrid in the next 24 hours. In actual operation, we will design fluctuations in renewable energy generation and user load demand, and there is a 10% error between the prediction and the actual situation. In order to reduce the impact of randomness, all experiments are independently run for 31 times, and the averaged results are listed in Table 4.

It is worth noting that the optimization results based on the MPC framework are significantly better than the optimization results of the day-ahead scheduling. This is because when optimizing the energy management of the microgrid based on the MPC framework, the algorithm can update the system's status according to the actual operating data in each period. On the contrary, since there is no real-time feedback mechanism in the day-ahead scheduling method, its optimal

Input: maximum generations Max Gen, population size N
Output: nondominated set PF, knee point kc
(1) $kc \longleftarrow \phi$ and $n\_nc \longleftarrow 0$
(2) PS $\longleftarrow$ Initialization (N)
(3) Costs $\longleftarrow$ Cal Obj Value (PS)
(4) **While** $it_{cur} \leq it_{max}$
(5) $OS_1 \longleftarrow$ Crossover (PS, kc)
(6) $OS_2 \longleftarrow$ Mutation (PS)
(7) joint S = PS $\cup$ OS_1 $\cup$ OS_2
(8) EF $\longleftarrow$ Cal Env Fitness (joint S, kc)
(9) joint $S \longleftarrow$ Non dominated Sort (joint S)
(10) PS $\longleftarrow$ Env Selection (EF, joint S)
(11) $kc \longleftarrow$ Knee Detection (PS)
(12) $it_{cur} = it_{cur} + 1$
(13) **End While**

Algorithm 1: General framework.

Table 2: Parameter values of the microgrid system in this research.

T	k	Δt	P_{grid}^{min} P_{grid}^{max}	E_{bess}^{cur}	E_{bess}^{min} E_{bess}^{max}	P_{bess}^{min} P_{bess}^{max}	ε_{bess}	η_{bess}
24 hours	12, 18, 24 hours	1 hour	0, 1000 kW	500 kWh	0, 2000 kWh	−200, 200 kW	0.2 kWh	95%

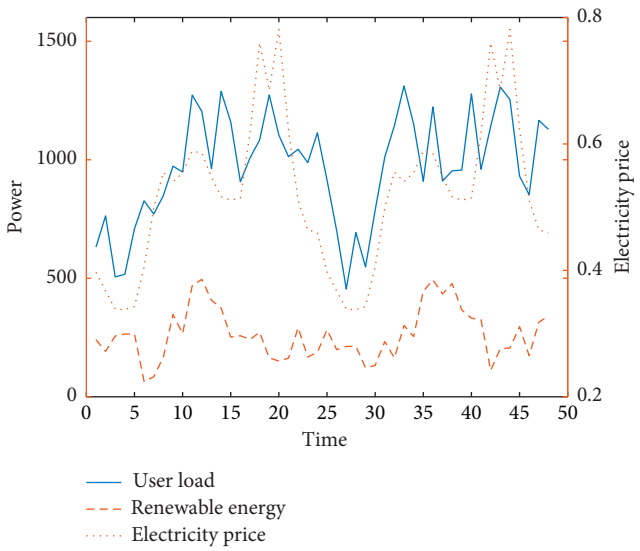


Figure 4: Schematic diagram of microgrid user load, renewable energy generation, and time-of-use electricity price.

results are poor. It is foreseeable that as the optimization period increases, the data prediction errors generated by the day-ahead scheduling will gradually accumulate, resulting in the continuous deterioration of the optimization effect.

4.4. Comparison of Results of Different Algorithms. The KBEMO algorithm proposed in this paper can utilize knee as the optimal solution for application after the optimization in each stage, without the need for decision-makers to give preference information. For the traditional method, the decision-maker needs to give a set of weight vectors in advance, after each optimization, and use the weighted sum (WS) method to convert the objective function vector into a scalar value, thereby selecting the best solution. To compare the efficiency and performance of the WS method and the KBEMO method, we use the knee obtained by the KBEMO algorithm and the optimal solution obtained by the WS method under the MPC framework to solve the dynamic microgrid energy management problem. For the WS method, since the dimension of the objective function involved in this problem is inconsistent, after each run, the objective function value is first normalized. Since it is impossible to determine which weight can obtain the best performance, we set up multiple different weight vectors. In order to reduce the impact of randomness, all experiments were run 31 times independently, and the average of the results was taken as the final result.

Figure 7 shows the results that are closest to the average running result based on different weight vectors and the KBEMO algorithm. This result shows the running state of the charging and discharging power of the energy storage system and the interaction power with the power grid over time. It can be seen from the figure that both methods can make corresponding adjustments according to the time-of-use electricity price. When the electricity price is relatively high, the optimizer chooses to use the energy storage system for discharge to reduce the cost of purchasing electricity from the power grid; on the contrary, when the electricity price level is low, the optimizer chooses to charge the energy storage system with high power and purchase electricity from the power grid. This shows that the proposed microgrid energy management model can well describe the operating status of the microgrid.

It is worth mentioning that the running state of the knee solution is a good trade-off of the two objective functions. Solutions with $w = [0.1, 0.9]$ mainly consider to optimize

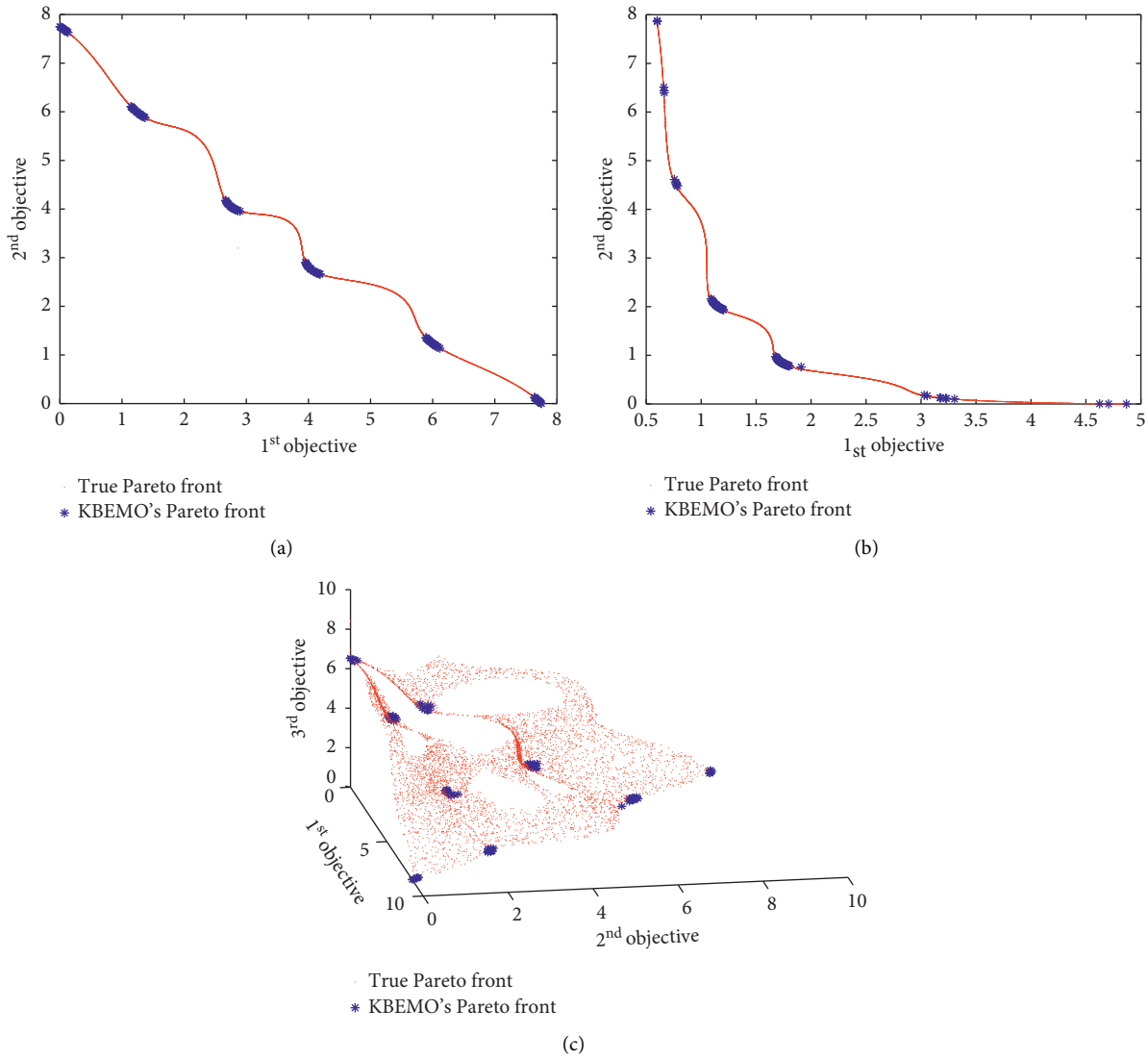


Figure 5: Pareto fronts obtained by KBEMO on knee test problems: (a) DEB2DK; (b) DO2DK; (c) DEB3DK.

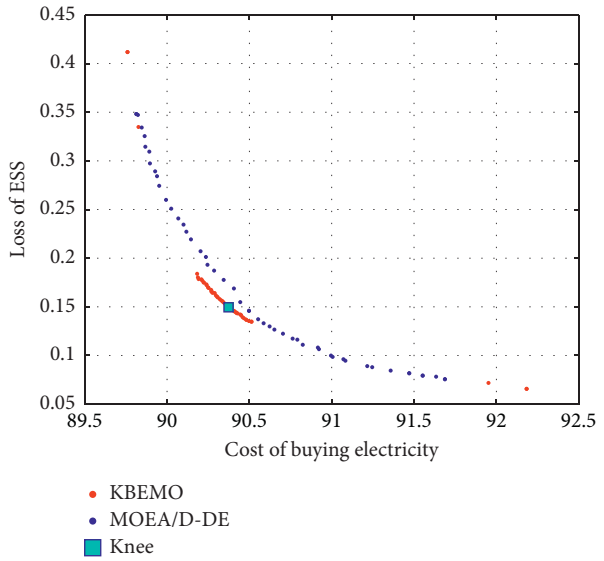


Figure 6: Pareto fronts obtained by MOEA/D-DE and KBEMO.

Table 3: Algorithm running parameters.

The iterations of one time	Iterations	Cross probability	Mutation probability	Population size
100	40	0.7	0.3	80

Table 4: Results of microgrid.

Method	Actual cost of buying electricity	Loss of the energy storage system
DA	92.7634$	0.7856
MPC	89.6854$	0.5598

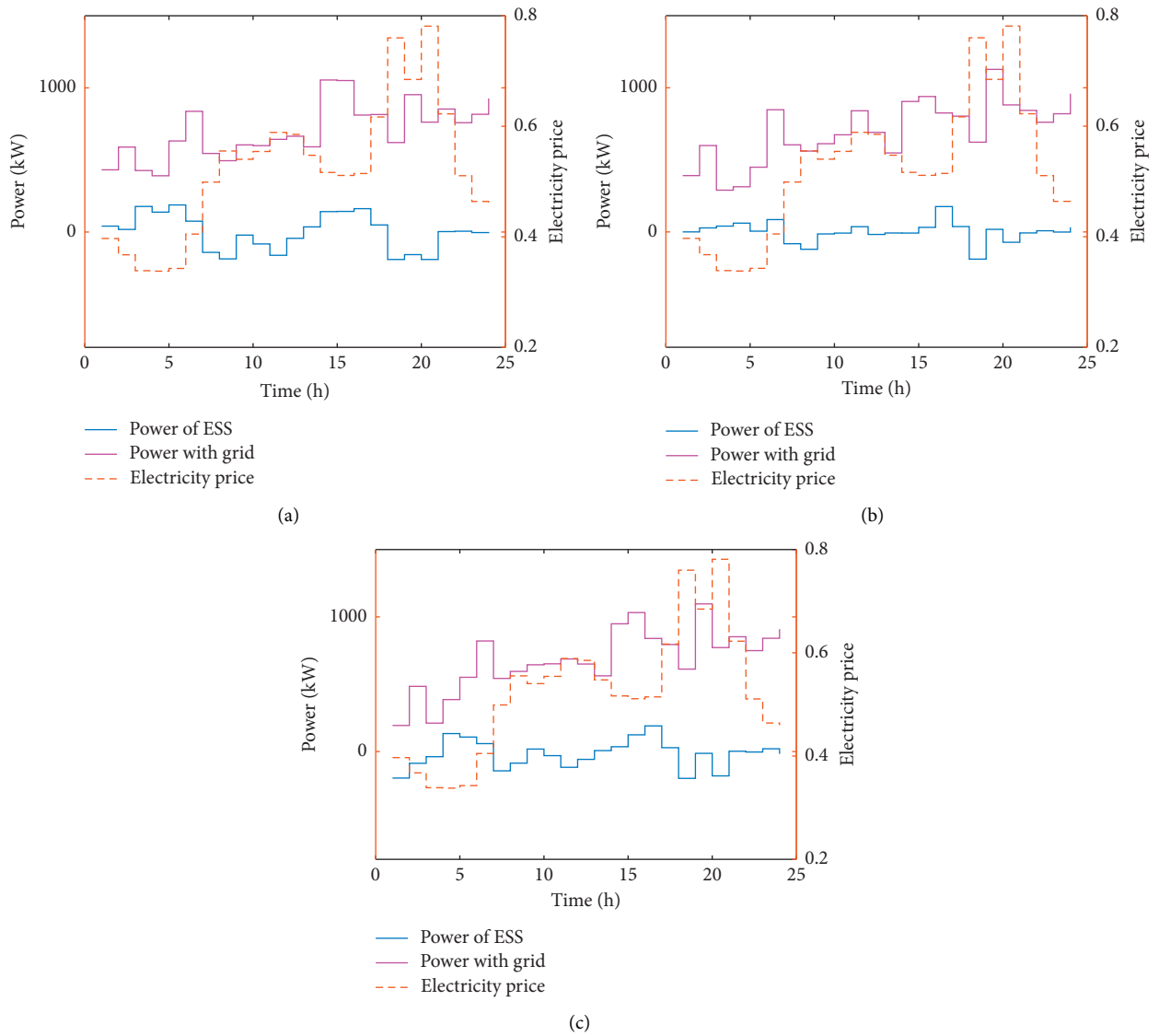


Figure 7: Operation states of different components: (a) $w = [0.1, 0.9]$; (b) $w = [0.9, 0.1]$; (c) knee.

the 2nd objective function, and the loss of energy storage system is good but results in high electricity buying cost. On the contrary, the solution with $w = [0.9, 0.1]$ will consider the 1st objective function more.

It can be seen from Figure 8 that choosing solutions by different weights at each stage will result in the final result being more biased towards the objective function. Therefore, the WS method can reflect the preference information of the decision-maker to a certain extent. However, the results of most weight vectors under the WS method are inferior to the KBEMO algorithm. In other words, the objective function value obtained by the KBEMO algorithm dominates the objective function value obtained by the WS method under most weight values. On the one hand, during the operation

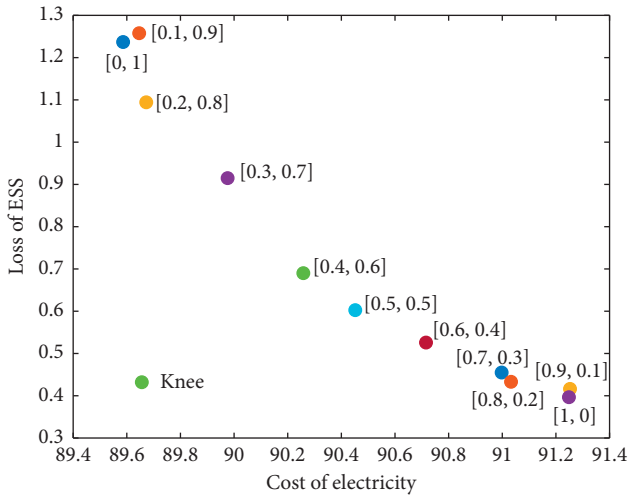


Figure 8: Solutions distribution of WS and KBEMO.

of the MPC framework, the KBEMO algorithm can obtain the most balanced solution every time, so the final solution obtained by this method can well balance the two objectives. The WS method selects the solution with a specific weight vector every time, and the trade-off between the two goals is poor, so the quality of the obtained solution is not high. On the other hand, because the KB algorithm focuses on searching the knee area, its convergence performance is better than the classic multiobjective optimization algorithm.

5. Conclusion

Microgrid plays an important role in making use of renewable energy and restraining fluctuation of the power grid supply. With the increase in the dimensions of the problem to be considered, the role of multiobjective optimization algorithms in the energy management of microgrids is becoming more and more important. Although the multiobjective optimization algorithm can give the Pareto optimal front of the problem, the decision-maker still needs to choose one solution from many solutions to apply. This will put pressure on decision-makers and consume computing resources to search for areas that decision-makers may not be interested in. Knee, as the largest marginal utility point on the Pareto front, is considered to be a more popular solution for decision-makers.

Aiming at the situation that the optimal solution needs to be continuously determined in the model predictive control, this paper proposes an evolutionary multiobjective optimization algorithm based on the knee, which can find the knee region of the Pareto front during the evolution process. The algorithm is used to optimize the energy management problem of the microgrid, and the knee is applied as the optimal solution in each period. Experiments have proved that this method can effectively obtain an optimized scheme for microgrid energy management and requires less computing resources.

Future research work includes the following: (1) fully considering the uncertainty of power generation and consumption and improving the multiobjective energy management method of microgrid combined with a model predictive control method. (2) Improving the knee-based EMO algorithm so that the algorithm can obtain the knee area of the problem more accurately and reliably and improve the computational efficiency of the algorithm. For now, the KBEMO is based on NSGA-II, which is proposed several years ago. Some recent algorithms with better performance can be considered [24–26]. (3) Consider applying the knee-based EMO algorithm to the multi-microgrid group, which provides new ideas for the efficient solution of the multi-microgrid system.

Data Availability

All data are included in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61773390), the Hunan Youth Elite Program (2018RS3081), the scientific key research project of National University of Defense Technology (ZK18-02-09 and ZZKY-ZX-11-04), and the Key Project of 193-A11-101-03-01.

References

[1] R. Lasseter and P. Paigi, "Microgrid: a conceptual solution," in *Proceedings of the 2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No. 04CH37551)*, Aachen, Germany, June 2004.

[2] F. Wang, J. L. Duarte, and M. A. M. Hendrix, "Grid-interfacing converter systems with enhanced voltage quality for microgrid application-concept and implementation," *IEEE Transactions on Power Electronics*, vol. 26, no. 12, pp. 3501–3513, 2011.

[3] M. Sechilariu, B. Wang, and F. Locment, "Building-integrated microgrid: advanced local energy management for forthcoming smart power grid communication," *Energy and Buildings*, vol. 59, pp. 236–243, 2013.

[4] W. Li, R. Wang, T. Zhang, M. Ming, and H. Lei, "Multi-scenario microgrid optimization using an evolutionary multi-objective algorithm," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100570, 2019.

[5] G. Liu, Y. Xu, and K. Tomsovic, "Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 227–237, 2015.

[6] Y. Zhang, L. Fu, W. Zhu, X. Bao, and C. Liu, "Robust model predictive control for optimal energy management of island microgrids with uncertainties," *Energy*, vol. 164, pp. 1229–1241, 2018.

[7] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[8] S. Phrakonkham, J.-Y. Le Chenadec, D. Diallo, G. Remy, and C. Marchand, "Reviews on micro-grid configuration and

dedicated hybrid system optimization software tools: application to Laos," *Engineering Journal*, vol. 14, no. 3, pp. 15–34, 2010.

[9] W. Li, R. Wang, T. Zhang, M. Ming, and K. Li, "Reinvestigation of evolutionary many-objective optimization: focus on the pareto knee front," *Information Sciences*, vol. 522, pp. 193–213, 2020.

[10] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: a comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

[11] W. Li, T. Zhang, R. Wang, B. Wang, Y. Song, and X. Li, "A knee-point driven multi-objective evolutionary algorithm for flexible job shop scheduling," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1716–1722, Xiamen, China, December 2019.

[12] Y. Kuo, T. Yang, and G.-W. Huang, "The use of grey relational analysis in solving multiple attribute decision-making problems," *Computers & Industrial Engineering*, vol. 55, no. 1, pp. 80–93, 2008.

[13] M. Behzadian, S. Khanmohammadi Otaghsara, M. Yazdani, and J. Ignatius, "A state-of the-art survey of TOPSIS applications," *Expert Systems with Applications*, vol. 39, no. 17, pp. 13051–13069, 2012.

[14] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[15] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, pp. 722–731, Birmingham, UK, September 2004.

[16] I. Das, "On characterizing the "knee" of the Pareto curve based on normal-boundary intersection," *Structural Optimization*, vol. 18, no. 2-3, pp. 107–115, 1999.

[17] W.-Y. Chiu, G. G. Yen, and T.-K. Juan, "Minimum Manhattan distance approach to multiple criteria decision making in multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 972–985, 2016.

[18] G. Yu, Y. Jin, and M. Olhofer, "An a priori knee identification multi-objective evolutionary algorithm based on α-dominance," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 241-242, Prague, Czech Republic, July 2019.

[19] L. Rachmawati and D. Srinivasan, "Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 810–824, 2009.

[20] G. Yu, Y. Jin, and M. Olhofer, "Benchmark problems and performance indicators for search of knee points in multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3531–3544, 2020.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[22] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, pp. 105–145, Springer, Berlin, Germany, 2006.

[23] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[24] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[25] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[26] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

Hindawi
Complexity
Volume 2020, Article ID 8864239, 11 pages
https://doi.org/10.1155/2020/8864239

Research Article

Improved Density Peaks Clustering Based on Natural Neighbor Expanded Group

Lin Ding,[1] **Weihong Xu**,[1,2] **and Yuantao Chen**[1]

[1]*School of Computer and Communication Engineering and Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, Hunan 410114, China*
[2]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China*

Correspondence should be addressed to Yuantao Chen; chenyt@csust.edu.cn

Received 23 August 2020; Revised 13 September 2020; Accepted 1 October 2020; Published 23 October 2020

Academic Editor: Shi Cheng

Density peaks clustering (DPC) is an advanced clustering technique due to its multiple advantages of efficiently determining cluster centers, fewer arguments, no iterations, no border noise, etc. However, it does suffer from the following defects: (1) difficult to determine a suitable value of its crucial cutoff distance parameter, (2) the local density metric is too simple to find out the proper center(s) of the sparse cluster(s), and (3) it is not robust that parts of prominent density peaks are remotely assigned. This paper proposes improved density peaks clustering based on natural neighbor expanded group (DPC-NNEG). The cores of the proposed algorithm contain two parts: (1) define natural neighbor expanded (NNE) and natural neighbor expanded group (NNEG) and (2) divide all NNEGs into a goal number of sets as the final clustering result, according to the closeness degree of NNEGs. At the same time, the paper provides the measurement of the closeness degree. We compared the state of the art with our proposal in public datasets, including several complex and real datasets. Experiments show the effectiveness and robustness of the proposed algorithm.

1. Introduction

Clustering algorithm, usually as unsupervised learning, is a type of fundamental technique of machine learning [1]. It aims to divide a dataset into several subsets, which are also called categories, clusters, groups, etc, according to similarity, dissimilarity, or distance of samples. Hence, unlike supervised learning [2–18], clustering methods implement classification tasks without any prior knowledge and have been applied to image processing, pattern recognition, bioinformatics, data mining, the Internet of things, and other fields.

Due to flexibility and validity, various clustering algorithms have been proposed one after another. Jain classified these methods into partitioning-based, model-based, hierarchical-based, grid-based, and density-based approaches [19]. Partitioning methods aim for grouping the dataset into a preset number of clusters via an iterative process. *K*-means [20, 21] and Fuzzy *c*-means [22, 23] are two famous partitioning-based clusterings. Although they are simple to understand and easy to implement, *K*-means is extremely sensitive to outliers and the selection of the initial cluster centers; besides, Fuzzy *c*-means approaches suffer from initial partition dependence [1]. Model-based clustering methods require one or more appropriate probability models to represent the dataset and often use the expectation-maximization approach to maximize the likelihood function [24]. Hierarchical-based approaches [25–28] partition the dataset into several categories using two opposite ways: top-down or bottom-up approach [23]. The first one considers the whole dataset as a cluster and split it into a suitable number of subclusters. Another regards each sample as a cluster and then merging these atomic clusters into more and more massive clusters. However, the effectiveness of hierarchical clustering algorithms depends on the type of distance measurement chosen for the clusters. Grid-based [29] and density-based [30, 31] approaches automatically determine the number of categories using suitable and preset

parameters such as epsilon, min-pts, or others. While it is necessary to take a mass of argument adjustments to obtain optimal clustering results, these two types of algorithms generate noise at the cluster borders.

To overcome the above shortcomings, recently, density peaks clustering [32] is proposed and based on the assumption that cluster centers are relatively denser and are far from each other. Using a suitable value of cutoff distance (namely, dc, the only parameter of DPC), this approach manually selects the appropriate center of each cluster from a decision graph. It then assigns each of the remaining elements to the nearest denser point (NDP) that is the nearest one of neighbors possessing bigger density than the assigned sample. It has many advantages, including higher efficiency in finding cluster centers, fewer parameters, no iterations, and no noise around the cluster border. However, the algorithm is still affected by the following defects:

(1) It is challenging to determine suitable dc. It must also be mentioned that the original DPC algorithm does not cover a reliable and specific method to ensure proper dc. Besides, this was demonstrated in several studies [33, 34] that DPC is sensitive to its parameter, and even when being normalized or using the relative percentage method, a small change in dc will still cause a conspicuous fluctuation in the result.

(2) The formula of local density is too simple to find out suitable center(s) of the sparse cluster(s) and is only useful in datasets with balanced density [33]. As shown in Figure 1(a), the Jain dataset has two clusters: the upper one is sparse and the lower one is denser. However, DPC overlooks the center of the upper cluster, instead of a prominent density peak of the lower cluster.

(3) Its assignment strategy is not robust [35]. Each point is assigned to its NDP, which results in some prominent density peaks (PDP) that are relatively bigger on density and δ_i value but not cluster centers are mistakenly attributed to a denser superordinate but are far away from each other. Accordingly, the subordinates of the incorrect-assigned PDP are portioned to an incorrect group. Figure 1(b) shows that we manually modify the center to the densest point of the upper cluster. However, the prominent local peak of the top cluster is assigned to its NDP belonging to the lower cluster, which leads to the incorrect assignment of its subordinates. And there is a distinct gap between the assignment path.

To improve the performance of DPC and inspired by the idea of natural neighbor (NN) [36], we propose an improved density peaks clustering based on natural neighbor expanded group. The main innovations and improvements in our algorithm are as follows:

(1) Define natural neighbor expanded and natural neighbor expanded group based on the well-known K-nearest neighbor method and its optimal version named natural neighbor. The concept of natural neighbor expanded is to absorb those close neighbors overlooked by the NN method. And NNEG is able to overcome the shortcoming of the remote assignment of PDP and mine the potential structure of data.

(2) Provide a density metric formula based on NNE. With the aid of NNE, the new measurement adaptively calculates the local density for each sample without any arguments, unlike one of the original DPC.

(3) Propose the measurement of the closeness degree of NNEGs that based on the mutual and pairwise neighbors which belonged to different NNEGs. Due to its application, all NNEGs are divided into the goal number of sets as the final clustering result.

(4) The time complexity is $O(K n \log n)$, where K is a constant, while the time complexity of all of the optimization algorithms and DPC is $O(n^2)$ [34].

The remainder of this paper comprises four sections. Section 2 describes the related works. Section 3 represents the DPC, NN method, and details of our algorithm. Section 4 presents the clustering results on our proposal and related works. In Section 5, we have a summary of the contributions and features of this paper.

2. Related Works

To improve the performance of the DPC algorithm, scholars proposed many optimization methods, as shown in Figure 2. Xie et al. modified the density metric formula using the K-nearest neighbor (KNN), which used the number of the nearest neighbors to replace dc. Besides, they devised an entirely new assignment scheme based on fuzzy weighted K-nearest neighbors (FKNN-DPC) [33]. Furthermore, this method is easier to determine the suitable value of parameter. Lotfi et al. proposed a technique called IDPC [37]. The algorithm sorts samples using the local density and then apportions the labels of centers to their KNN to develop cluster cores. Finally, IDPC implements a specific propagation strategy to attach the remaining points with labels. Guo et al. capitalized on the linear regression method to fit the decision values of DPC with a preset proper dc required (DPC-LRA) and then choose the instances above the fitting function as the centers [38]. Ding et al. proposed an algorithm based on the generalized extreme value distribution (GEV) to fit the DPC decision values in the descending order (DPC-GVE). To reduce the time complexity, they also represented a substitution method using Chebyshev inequality (DPC-CI) [39]. Ni et al. presented the definitions of density gap and the density path, as well as a new threshold [35]. Instead of the decision graph of DPC, the proper value of dc is determined by manually observing a summary graph incorporating the density gaps calculated by different dc. The method, named PPC, is able to reduce obviously the difficulty on threshold determination. Jiang et al. provided a novel density peaks clustering algorithm based on K-nearest neighbors (DPC-KNN) to overcome the issue of the assignment [40]. In this method, there are two sets for each sample i: the first one is S_i, which is composed of sample i and its KNN, while the second is H_i, which covers the data points possessing

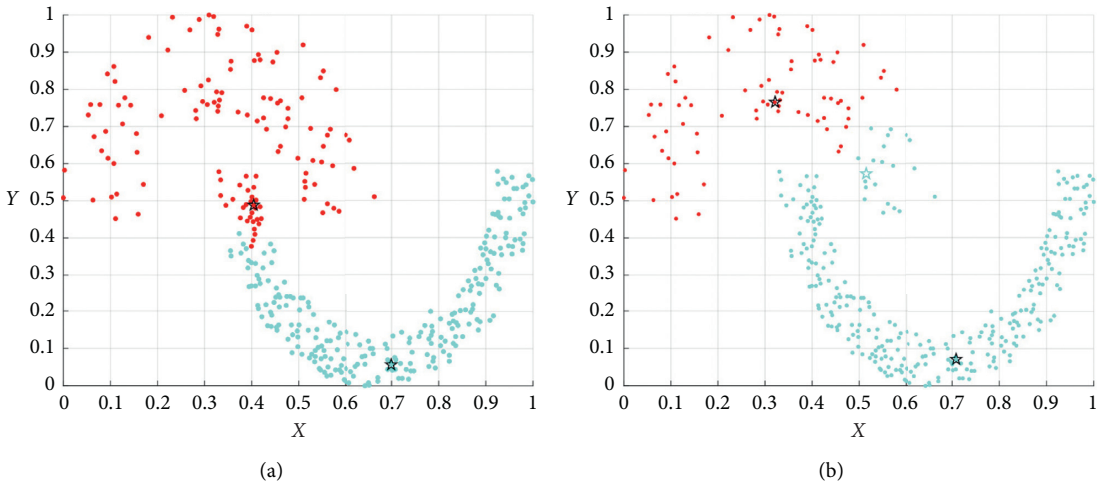


Figure 1: Clustering results of DPC on the Jain dataset. The diverse colors present different clusters, and the stars mark the cluster centers and prominent local peaks. (a) Clustering results of DPC on Jain. (b) Clustering results of DPC with modified center.

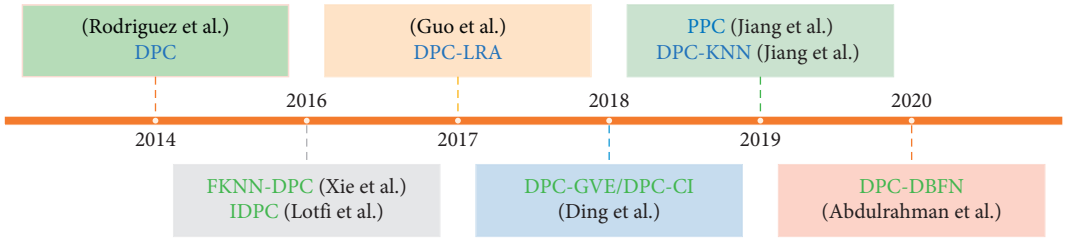


Figure 2: Improvement methods of DPC.

higher densities than sample i in the whole dataset. The cluster centers are determined via the decision graph of DPC, DPC-KNN assigns each remaindering sample to an element of H_i, who has the smallest distance from any member of S_i to any member of H_i. Lotfi et al. improve DPC using density backbone and fuzzy neighborhood (DPC-DBFN). They use a fuzzy kernel for improving the separability of clusters. DPC-DBFN uses a density-based KNN graph for labeling backbones and effectively assigns correct category labels to samples around the group borders to effectively cluster data with various shapes and densities [34].

However, FKNN-DPC, IDPC, DPC-KNN, PPC, and DPC-DBFN require manual operations. And a preset dc is necessary for DPC-LRA, DPC-GVE, and DPC-CI. Moreover, DPC and these algorithms require the time complexity of $O(n^2)$ [34].

3. Methods

This section aims to present the short versions of the original DPC algorithm and NN method and show a detailed description of our method.

3.1. The Original DPC Algorithm. DPC is the basis on which cluster centers are relatively denser and are distant from each other. For a given dataset $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}, i = 1, 2, \ldots, n$, cluster centers are manually picked from the decision graph, which is two-dimensional with δ_i as the ordinate and the local density as the abscissa. Local density is to measure the neighbor number and distances of each sample in its neighborhood, which is a crucial concept of DPC. The ordinate δ_i is the distance between the sample i and its nearest denser point. Since the centers have relative lager density, each of them must be far away from their NDP, namely, has an enormous value of δ_i. In the two-dimensional coordinate system, cluster centers simultaneously possess big values of δ_i and local density and appear in the upper right corner of the graph. To measure the local density of each element, the author provides two formulae expressed as equations (1) and (2). δ_i is calculated by equation (3):

$$\rho_i = \sum_j \aleph\left(d_{ij} - dc\right),$$
$$\aleph(\cdot) = \begin{cases} 1, & \cdot < 0, \\ 0, & \cdot \geq 0, \end{cases} \tag{1}$$

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{dc}\right)^2\right), \tag{2}$$

where d_{ij} is the distance between pairwise elements i and j, dc is the cutoff distance, the only argument of DPC. Therefore, The DPC algorithm inherits a defect, where Gaussian kernel is sensitive to bandwidth:

$$\delta_i = \begin{cases} \min_{j:\ \rho_i < \rho_j}\left(\mathrm{d}_{ij}\right), & \text{if } \exists j \text{ s.t. } \rho_i < \rho_j, \\ \max_j\left(\mathrm{d}_{ij}\right), & \text{otherwise,} \end{cases} \tag{3}$$

As shown in equation (3), δ_i is the minimum distance between elements i and j whose density is higher than i. For i with the highest density, its δ_i is the maximum distance between i and j. After the cluster centers have been found, each remaining point is assigned to the same cluster as its nearest neighbor of higher density.

3.2. Natural Neighbor Method. K-nearest neighbor is a popular method in machine learning to complete the tasks of classification and clustering. However, the crucial argument K is preset manually. And natural neighbor is an adaptive method to find the relative near neighbors of each sample. The basic idea of NN is that samples of the dense regions have more neighbors; data points of the sparse area have relatively fewer neighbors; the outliers only have a few or no natural neighbors.

In the dataset X, the authors assume that s_{ij} is the similarity between two points x_i and x_j. With the help of comparing the similarity, let find $\mathrm{KNN}(x_i, n)$ denote the function of KNN searching which returns the r^{th} nearest neighbor of the point x_i, $\mathrm{KNN}_r(x_i)$ is a subset of X, and it is defined as follows:

$$\mathrm{KNN}_r(x_i) = \bigcup_{n=1}^{r} \{\text{find KNN}(x_i, n)\}. \tag{4}$$

Definition 1. (natural neighbor). Natural neighbor of x_i is defined as

$$x_j \in \mathrm{NN}(x_i) \Leftrightarrow \left(x_i \in \mathrm{KNN}_\lambda\left(x_j\right) \wedge x_j \in \mathrm{KNN}_\lambda(x_i)\right). \tag{5}$$

Definition 2. (natural neighbor eigenvalue). When the algorithm reaches the Stable Searching State, Natural Neighbor Eigenvalue (NaNE) λ is equal to the searching round r:

$$\lambda \triangleq r_{r\in N}\left\{r \middle| (\forall x_i)\left(\exists x_j\right)(r \in N) \wedge \left(x_i \neq x_j\right) \longrightarrow \left(x_i \in \mathrm{KNN}_r\left(x_j\right) \wedge x_j \in \mathrm{KNN}_r(x_i)\right)\right\}. \tag{6}$$

3.3. The Proposed Method. In this section, the improved density peaks clustering based on natural neighbor expanded group is presented. Our method includes three major steps, including (1) calculating the local density of each sample according to the formula proposed, (2) determining natural neighbor expanded groups, and (3) grouping NNEGs into several sets as the final clustering result. The details of these steps are described in the remaining part of this section. To realize the above processing, we define the concept of natural neighbor expanded and then provide a straightforward but useful formula for local density. Besides, the definition of the natural neighbor expanded group is to reveal the structure of the dataset and divide the dataset into several local groups. For ensuring the grouping of NNEGs accuracy, we propose a measurement of closeness degree. And more details are presented in the rest content of this section.

3.3.1. Basic Concepts. The NN method only considers the relationship of mutual neighbors and overlooks the impact of distance between samples. And to fit the density metric and the searching of density peaks, we propose the concept of Natural Neighbor Expanded.

Definition 3. (natural neighbor expanded). Natural Neighbor Expanded is defined as the following equation:

$$\mathrm{NNE}(x_i) = \mathrm{KNN}_{2K_i}(x_i), \tag{7}$$

where we assume that the number of NN of x_i is $|\mathrm{NN}_i|$ and the $|\mathrm{NN}_i|^{\mathrm{th}}$ $\mathrm{NN}(x_i)$ is the $K_i{}^{\mathrm{th}}$ $\mathrm{KNN}_r(x_i)$. Hence, $K_i <= r$. As shown in Figure 3, sample 1 is not the NN of sample 8, since it does not belong to the $\mathrm{KNN}_6(8)$. However, sample 1 is closer to sample 8 than 14. Hence, for calculating the density more wholly and accurately, we expand the natural neighborhood of sample 8 to include samples 1, 2, and 7.

Natural Neighbor is the set of close neighbors. Still, as shown in equation (2), the local density formula measures not only the close neighbors whose distances to sample i are smaller than dc but also the rest samples of whole datasets. In the latter part, the distance to sample i being approximate to dc and the corresponding sample also impacts the density of i. Therefore, $2K_i$ of equation (7) is to cover the more secondary-adjacent samples beside the close neighbors. And the new local density formula based on NNE is shown as

$$\rho_i = \sum_{j\in \mathrm{NNE}(x_i)} \frac{\max(\mathrm{distNNE}) - \mathrm{d}_{ij}}{\max(\mathrm{distNNE}) - \min(\mathrm{distNNE})}, \tag{8}$$

where $\mathrm{NNE} = \cup_{i=1}^{n}\mathrm{NNE}(x_i)$, $\mathrm{distNNE} = \cup_{i=1}^{n}\mathrm{distNNE}(x_i)$, and $\mathrm{distNNE}(x_i)$ is the set of the distances of x_i to all of the elements in $\mathrm{NNE}(x_i)$. Inspired by the famous K-means method, equation (8) considers each point as core and calculates the sum of distances of it to its NNE. And the smaller the distance sum is, the more likely it is to be the local center.

Equation (2) maps the distances to similarities using the Gaussian kernel and calculates the accumulation sum of

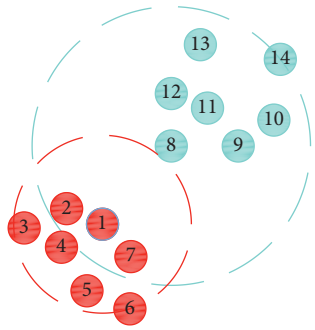


Figure 3: A schematic diagram with parts of samples and its r equals 6. Sample 1 has six NNEs, including samples 2, 3, 4, 5, 6, and 7; sample 8 has six NNEs, including samples 9, 10, 11, 12, 13, and 14.

similarities linking to x_i as ρ_i. Hence, equation (2) based on Gaussian kernel can resist the interference of outliers that possess vast distances to x_i. However, the equation covers too many negligible samples that have the distances to sample i much bigger than dc because their contribution to density is tiny through the mapping of the Gaussian kernel. Moreover, it brings the original DPC to the time complexity of $O(n^2)$.

In contrast, our formula only considers NNE. It, therefore, also gets rid of the passive impact of outliers since they usually are distant from its nearest point and are not in any NNE(s) of other(s), at the same time, reduce the computational complexity. And unlike the Gaussian kernel mapping, equation (8) retains the original information of data, does not require any arguments, and avoids the sensitivity caused by dc.

Definition 4. (natural neighbor expanded group). Natural Neighbor Expanded Group consists of a prominent density peak and its subordinates.

In our method, each point is assigned to the nearest denser point of its NNE. The assignment process is stored in a list: the index numbers represent the samples in the given dataset, respectively; each unit stores the index number of its superordinate one, and if the density of a sample is bigger than all of its NDP, the related unit saves 0. Namely, zero samples are prominent density peaks. The assignment divides the dataset into several NNEGs, adaptively.

Essentially, NNEGs reveal the potential structure of the dataset analyzed and are relatively tighter subcluster and local groups in the cluster of the Ground Truth. Due to the application of NNEG, each sample only points to a neighbor, and our method could avoid the long-distance assignment of the PDP.

As shown in Figure 4, after NNEGs are determined, our method only needs to merge such local groups into the goal number of clusters and hence remove the operation of the center selection from the decision graph, which overcomes the mentioned issue of the density metric of DPC. To clarify the close relationship between NNEGs, we proposed the concept of the adjacent group graph.

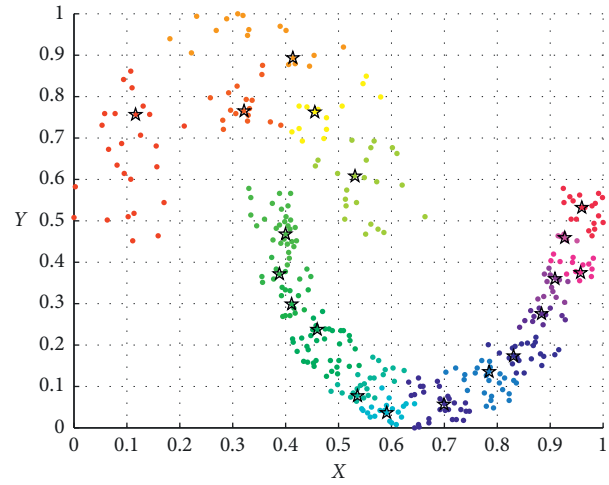


Figure 4: NNEGs of the Jain dataset. The diverse colors present different NNEGs, and the stars mark the zero samples.

Definition 5. (adjacent group graph). $\mathrm{AGG} = G(V, E)$, where $V = \{v_1, v_2, \ldots, v_k\}$ is a set of NNEGs, $E = \{E(v_t, v_\tau) | t \neq \tau,\ v_t,\ v_\tau \in V\}$, and $E(v_t, v_\tau)$ is a set of edges linked to NNEGs v_t and v_τ, and subject to

$$E(v_t, v_\tau) = \{e(x_i, x_j) | (x_i \in (v_t \wedge \mathrm{NNE}(x_j))) \wedge (x_j \in (v_\tau \wedge \mathrm{NNE}(x_i)))\}. \tag{9}$$

Adjacent Group Graph usually is a multigraph, since there could be several $e(x_i, x_j)$ between v_t and v_r. And the more the edges are, the closer the two groups are. Obviously, in Figure 4, there are no edges between the upper and the lower clusters. Moreover, the degree of closeness (DC) of the neighboring pairwise NNEGs is calculated by

$$\mathrm{DC}(v_t, v_\tau) = \sum_{e(x_i, x_j) \in E(v_t, v_\tau)} w_i w_j \frac{\max(\mathrm{distNNE}) - d_{ij}}{\max(\mathrm{distNNE}) - \min(\mathrm{distNNE})}, \tag{10}$$

where $w_i = (|\mathrm{NNE}(x_i) \cap v_t|)/(|\mathrm{NNE}(x_i)|)$ and $w_j = (|\mathrm{NNE}(x_j) \cap v_r|)/(|\mathrm{NNE}(x_j)|)$. As shown in equation (10), the formula of closeness degree is constituted with two parts: the weight and the similarity normalized. It is based on an assumption where the more compact the endpoints and their respective NNEGs are, the more reliable the edge is. w_i represents the compactness between the sample x_i and the group v_t, viz., the bigger number of intersected elements of $\mathrm{NNEG}(x_i)$ and v_t means the relationship between them is intenser. To ensure $w_i \in [0, 1]$, the number of the elements intersected divided by $|\mathrm{NNE}(x_i)|$.

3.3.2. The Specific Processing

Inputs: dataset X, the goal number of clusters.

Output: the clustering result.

Step 1: Create a k-d tree. Search NNE for each sample using the k-d tree.

Step 2: Calculate local density according to equation (8).

Step 3: Determine NNEG according to Definition 4.

Step 4: Generate the Adjacent Group Graph as in Definition 5, and find all edges of each pairwise NNEGs as equation (9).

Step 5: Calculate the degree of closeness, according to equation (10).

Step 6: Break up the original cluster containing all NNEGs into the goal number of sets, according to the closeness degree.

To clarify Step 6 in detail, we present an example in Table 1. As shown in Table 1 (A), there are five NNEGs in a dataset. And the closeness degrees of adjacent pairwise NNEGs are recorded. Assume the goal number is 2. Our method considers the whole dataset as a cluster, since $\mathrm{DC}(v_1, v_2), \mathrm{DC}(v_2, v_3), \mathrm{DC}(v_3, v_4), \mathrm{DC}(v_3, v_5), \mathrm{DC}(v_4, v_5) > 0$. We force the minimum $\mathrm{DC}(v_2, v_3) = 0$ as shown in Table 1 (B), which means those NNEGs are split into two parts: $\{v_1, v_2\}$ and $\{v_3, v_4, v_5\}$, i.e., split is a for-loop operation which let the minimum DC = 0 until the cluster number equals to the goal one.

And more details are as shown in the pseudocode. In the 6th line, AGG is a matrix where each row and each column correspond to one of NNEGs. In the 16th line, inspired by the Top-down hierarchical clustering, we consider the whole dataset as a cluster containing all NNEGs and break the weakest $E(v_t, v_\tau)$ in the AGG until the cluster number equals the goal, which corresponds to the process, Table 1 (A) and (B).

3.3.3. Time Complexity Analyses. This section aims to analyze the computational complexity of our method, and suppose that the number of total samples in a dataset is n, the number of NNEG is equal to n_{NNEG}, the goal number of clusters is G, the NDP of sample i is the $n\mathrm{NDP}_i^{th}$ neighbor, and the biggest K_i equals $\widehat{K}$. (Algorithm 1).

The time complexity of creating a k-d tree is $O(n \log n)$ [41]. It is demonstrated that determining NN for all samples also requires the cost of $O(n \log n)$ [36]. And for finding NNE, we can record the K_i in the processing of searching NN. Hence, the searching NNE of a sample only needs to $2K_i$ times search operation, and its whole complexity for all samples is less than $O(2\widehat{K}n)$. Our local density metric is based on NNE, and it is not necessary to generate a distance matrix and only needs to $2K_i$ times plus operations for each sample. Therefore, it is required with at most $O(2\widehat{K}n)$ for the time cost to calculate local densities of all instances. For each sample, the method takes $n\mathrm{NDP}_i$ time search to find its NDP via the k-d tree in the round $2K_i$, and $n\mathrm{NDP}_i <= 2K_i$. In the process of generating each NNEG, we store the labels of its prominent density peak to a list where the first unit is any unallocated instance, and the end is an assigned one or prominent density peak. And the operation of storing labels of all samples only needs to the time cost of $O(n)$. And the cost required is $O(2\widehat{K}n)$ on dividing a dataset into G NNEGs. In equations (9) and (10), $e(x_i, x_j)$ is requested and determined via searching the NNE of each sample to find the neighbors having different labels. Thus, for all edges, it is equal to $O(2\widehat{K}n)$ for the magnitude of how many times the searching operation is performed. Furthermore, the time complexity of grouping in the last step must be less than G. Overall, we can conclude that the time complexity of the entire algorithm is $O(Kn \log n)$.

Table 1: An example of Step 6.

	v_1	v_2	v_3	v_4	v_5
v_1	—	4	0	0	0
v_2	4	—	1	0	0
v_3	0	1	—	3	5
v_4	0	0	3	—	4
v_5	0	0	5	4	—

(a)

	v_1	v_2	v_3	v_4	v_5
v_1	—	4	0	0	0
v_2	4	—	0	0	0
v_3	0	0	—	3	5
v_4	0	0	3	—	4
v_5	0	0	5	4	—

(b)

4. Results

In this section, several datasets are used to evaluate the performance of our method in comparison with some state-of-the-art techniques such as DPC-DBFN [34], DPC-KNN [40], IDPC [37], and FKNN-DPC [33]. The experiments are performed on a computer with a Windows 10, Intel (R) Core (TM) i7-8750H, 16 GB memory, and Matlab 2016b. The results represented are measured by several performance metrics, including Normalized Mutual Information (NMI) [42], Rand Index (RI) [43], and the Adjusted Rand Index (ARI) [44]. In this section, the similarity between points is measured using the Euclidean distance metric.

4.1. Datasets. In this paper, all tested datasets include three low-dimensional datasets and five high-dimensional datasets, which are public and from UCI. The two-dimensional datasets have different numbers of samples and different objective distributions. The DMI512 dataset containing 1024 elements with 512-dimensional features, which belonged to 16 Gaussian clusters sampled from a Gaussian distribution, is often used to test algorithm performance in high-dimensional space. Experiments of the four datasets, including Statlog (Shuttle), Abalone, Wine Quality, and Libras Movement, are applications of our method on Physical (the positioning of radiators in the Space Shuttle), Population Biology, Model Wine Preferences, and Hand Movement Recognition, respectively. And more details are presented in Table 2.

To reduce the influence of dimension weights and ensure the validity of the experimental comparison, we processed each dataset and normalized all dataset tested. The normalization formula is as follows:

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{11}$$

Require: Dataset $X = \{x_1, x_2, \ldots, x_n\}$, the goal number of clusters G
Ensure: The result of clustering: $C = \{C_1, C_2, \ldots, C_n\}$
(1) Create a k-d tree;
(2) Search the k-d tree;
(3) Determine NN according to [34], and record K_i, which means $\mathrm{NNE}(x_i)$ determined;
(4) Calculate local density ρ_i according to equation (8).;
(5) Assign each point to its NDP of its NNE to generate several NNEGs;
(6) Create a matrix AGG = (|NNEG|, |NNEG|);
(7) **for** $i = 1 : n$ **do**
(8) **for** $t = 1 : 2K_i$ **do**
(9) **if** the tth NNE and sample i belongs to different NNEGs do
(10) Calculate the closeness degree of this edge, referring to equation (10);
(11) Add the DC of this edge to the corresponding unit of AGG;
(12) **end if**
(13) **end for**
(14) **end for**
(15) **while** the number of clusters does not equal G do
(16) Store zero in the unit with the min value but greater than zero;
(17) Count the number of clusters;
(18) **end while**

Algorithm 1: DPC-NNEG.

Table 2: Detailed information on tested datasets.

Dataset	#Instance	#Attribute	#Cluster
Jain	373	2	7
Flame	240	2	2
Spiral	300	2	3
Statlog (shuttle)	58000	9	7
Abalone	4177	7	28
Wine quality	4898	11	7
DIM512	1024	512	16
Libras movement	360	90	15

where x_{ij} is the jth feature value of the ith sample, while $\max(x_j)$ and $\min(x_j)$ represent the maximum and minimum values of the jth feature, respectively.

4.2. Evaluation Measures. We tested our algorithm and several related works on the above datasets. For intuitive comparison, we chose RI, ARI, and NMI to measure the clustering results.

The RI formula is shown in

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{C_n^2}, \tag{12}$$

where TP indicates true positive, TN indicates real negative, and the denominator C_n^2 is the total number of sample pairs in a dataset consisting of n samples.

The ARI formula is shown in

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\mathrm{MAX}\{\mathrm{RI}\} - E[\mathrm{RI}]}, \tag{13}$$

where $E[\mathrm{RI}]$ represents the expectations of RI.

The NMI formula is shown in

$$\mathrm{NMI} = \frac{-2\mathrm{MI}(A,B)}{H(A) + H(B)}, \tag{14}$$

where $H(A) = \sum_{i=1}^{|A|} P(i)\log_2 P(i)$, $H(B) = \sum_{i=1}^{|B|} P(j)\log_2 P(j)$, $E[\mathrm{MI}(A,B)]$ represents the expectations of $\mathrm{MI}(A,B)$, and $\mathrm{MI}(A,B)$ is expressed as

$$\mathrm{MI}(A,B) = \sum_{i=1}^{|A|}\sum_{j=1}^{|B|} P(i,j)\log_2\frac{P(i,j)}{P(i)P(j)}, \tag{15}$$

where $P(i) = |A_i|/n$, $P(j) = |B_j|/n$, $P(i,j) = |A_i \cap B_j|/n$, $A = \{A_i | i = 1, 2, \ldots, |A|\}$, and $B = \{B_j | j = 1, 2, \ldots, |B|\}$. A and B represent two allocation methods for a dataset containing n elements, and A_i and B_j are clusters. In experimental verification, let A and B be the original labels and the clustering results of an algorithm, respectively. If the clustering results are as same as the real labels, the three metrics take the value of 1, and if the clustering results are entirely different from the labels, the values will be equal to 0.

4.3. Results. This section aims to show the detailed clustering results and evaluate the performance of different clustering algorithms on the various datasets. Tables 3–5 compare the performance of our method with DPC-DBFN, DPC-KNN, IDPC, and FKNN-DPC in terms of NMI, RI, and ARI measures, respectively. All these methods are using the KNN method, and the number of nearest neighbors (K) can be set from 1 to n. In these tables, the numbers in the parenthesis are the value of K, where the corresponding algorithm obtains the results represented, and boldface marks the best results.

The Jain dataset has 373 points and two clusters: the upper one and the lower one. As shown in Figure 5, DPC-NNEG divides the dataset into nineteen NNEGs and then successfully and efficiently groups them into two sets since there are no edges between the two clusters. Homoplastically, as shown in Figure 6, our algorithm

Table 3: Clustering results measured by NMI.

Dataset	DPC-KNN	IDPC	FKNN-DPC	DPC-DBFN	DPC-NNEG
Jain	1.0000 (9)	1.0000 (9)	1.0000 (10)	1.0000 (9)	1.0000
Flame	1.0000 (4)	1.0000 (7)	1.0000 (6)	1.0000 (9)	1.0000
Spiral	1.0000 (7)	1.0000 (5)	1.0000 (5)	1.0000 (4)	1.0000
Statlog (shuttle)	0.3734 (150)	0.1552 (90)	0.4226 (150)	0.5490 (7)	0.6101
Abalone	0.1780 (50)	0.1791 (758)	0.1828 (2)	0.1846 (2)	0.1852
Wine quality	0.0359 (28)	0.0339 (68)	0.0364 (244)	0.0701 (1)	0.0935
DIM512	1.0000 (10)	1.0000 (15)	1.0000 (8)	1.0000 (9)	1.0000
Libras movement	0.5287 (53)	0.5697 (5)	0.5607 (11)	0.5848 (6)	0.5855

Table 4: Clustering results measured by RI.

Dataset	DPC-KNN	IDPC	FKNN-DPC	DPC-DBFN	DPC-NNEG
Jain	1.0000 (9)	1.0000 (9)	1.0000 (10)	1.0000 (9)	1.0000
Flame	1.0000 (4)	1.0000 (7)	1.0000 (6)	1.0000 (9)	1.0000
Spiral	1.0000 (7)	1.0000 (5)	1.0000 (5)	1.0000 (4)	1.0000
Statlog (shuttle)	0.6780 (150)	0.4180 (90)	0.7274 (150)	0.7512 (7)	0.7814
Abalone	0.8042 (14)	0.8236 (60)	0.7635 (5)	0.8354 (500)	0.8428
Wine quality	0.6277 (52)	0.6063 (35)	0.5107 (3)	0.5578 (1)	0.5751
DIM512	1.0000 (10)	1.0000 (15)	1.0000 (8)	1.0000 (9)	1.0000
Libras movement	0.8839 (4)	0.9089 (9)	0.8995 (10)	0.8945 (6)	0.9187

Table 5: Clustering results measured by ARI.

Dataset	DPC-KNN	IDPC	FKNN-DPC	DPC-DBFN	DPC-NNEG
Jain	1.0000 (9)	1.0000 (9)	1.0000 (10)	1.0000 (9)	1.0000
Flame	1.0000 (4)	1.0000 (7)	1.0000 (6)	1.0000 (9)	1.0000
Spiral	1.0000 (7)	1.0000 (5)	1.0000 (5)	1.0000 (4)	1.0000
Statlog (shuttle)	0.3396 (150)	0.1049 (90)	0.3197 (150)	0.3584 (7)	0.5688
Abalone	0.0567 (3)	0.0589 (10)	0.0553 (7)	0.0654 (2)	0.0657
Wine quality	0.0275 (25)	0.0356 (64)	0.0214 (3)	0.0575 (1)	0.0511
DIM512	1.0000 (10)	1.0000 (15)	1.0000 (8)	1.0000 (9)	1.0000
Libras movement	0.2492 (19)	0.3337 (9)	0.2846 (11)	0.3130 (6)	0.4862

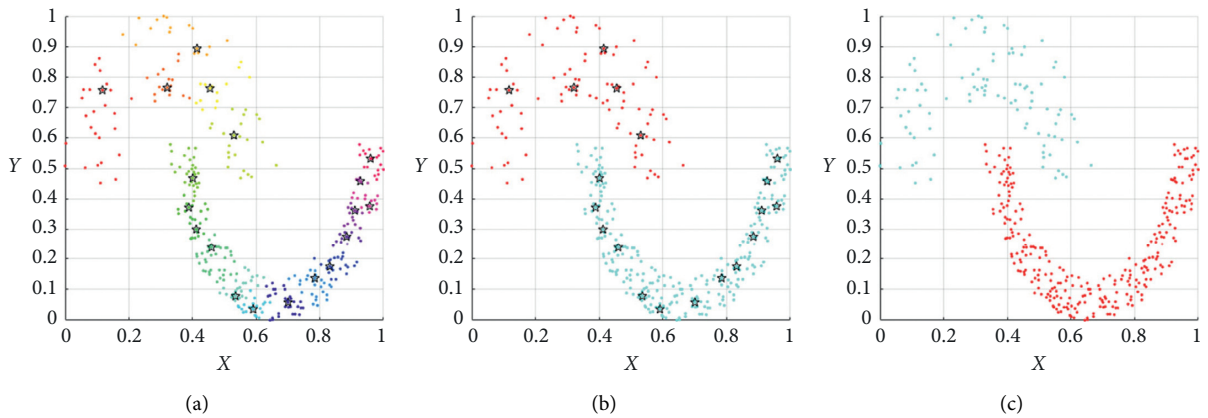


Figure 5: The clustering results of the Jain dataset. The diverse colors present different NNEGs and clusters, and the stars mark the zero samples. (a) NNEGs of Jain. (b) DPC-NNEG on Jain. (c) Ground truth.

divides the Spiral dataset into several local groups and subsequently merges all NNEGs accurately into the goal number of clusters.

Unlike Jain and Spiral, as shown in Figure 7, the Flame dataset containing 240 data points has no clear gap between the two adjacent clusters. Hence, it is more sensitive to the value of dc of the DPC algorithm because a tiny change in dc will cause the border point is assigned to another cluster. However, our method not only partitions all samples into eight NNEGs but also measures the

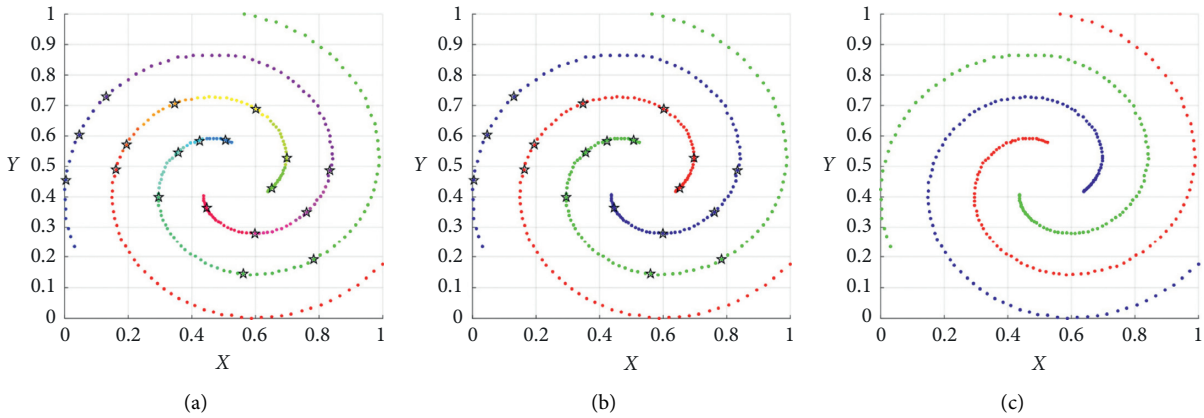


Figure 6: The clustering results of the Spiral dataset. The diverse colors present different NNEGs, and the stars mark the zero samples. (a) NNEGs of spiral. (b) DPC-NNEG on spiral. (c) Ground truth.

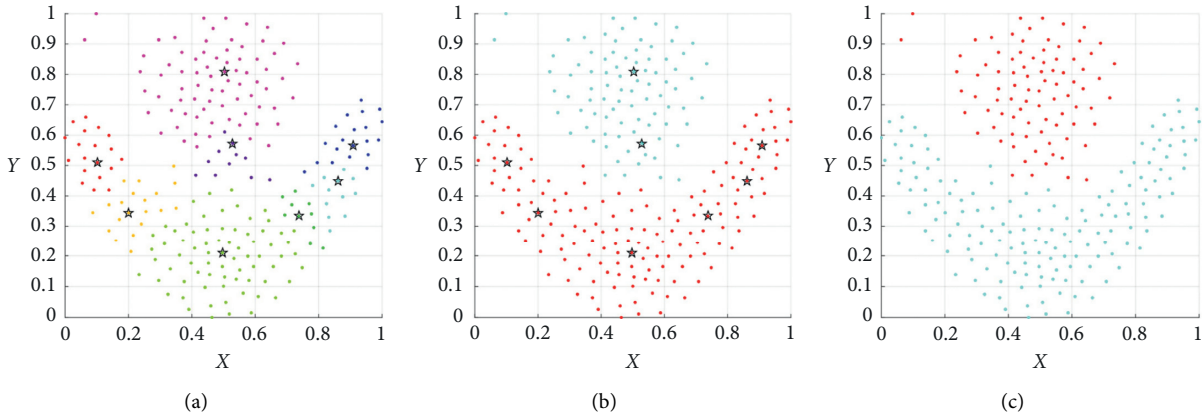


Figure 7: The clustering results of the Flame dataset. The diverse colors present different NNEGs, and the stars mark the zero samples. (a) NNEGs of flame. (b) DPC-NNEG on spiral. (c) Ground truth.

tightness between different groups accurately, which realizes the correct grouping of those local groups. And Figure 7 shows that the clustering result of Flame by DPC-NNEG is consonant with the Ground Truth.

As shown in Tables 3–5, there is no difference in performance among our algorithm, DPC-DBFN, DPC-KNN, IDPC, and FKNN-DPC in three two-dimensional datasets. However, as shown in Table 2, the clustering results of more complex high-dimensional datasets show the outperformance of our method: DPC-NNEG gains the best marks measured by NMI in all datasets. For example, the results of DPC-NNEG in the Statlog (Shuttle), Abalone, Wine Quality, DIM512, and Libras Movement datasets are 0.6101, 0.1852, 0.0935, 1.0000, and 0.5855, respectively. Moreover, its improvements to the second-best method (in %) for Statlog (Shuttle), Abalone, Wine Quality, and Libras Movement datasets are respectively 11.13, 0.32, 33.38, and 0.12.

Tables 4 and 5 show similar results, respectively, measured by RI and ARI. These results also demonstrate that the proposed method, in most cases, obtains the biggest values of NMI except the Wine Quality dataset. Hence, based on these results, it can be concluded that DPC-NNEG has given an overall excellent performance in clustering.

5. Conclusions and Future Works

This paper proposed an efficient clustering algorithm called DPC-NNEG, which can easily split a dataset into local groups and then merge those groups into the goal number of clusters with various densities, shapes, and sizes. The proposed method aims at clustering the data by three major steps: calculating the local density of each sample, identifying natural neighbor expanded groups, and merge those groups into clusters. The first step utilizes the natural neighbor method in the local density calculation. And it is entirely different from the formula of the original DPC and could avoid the impact of outliners and reduce the sensitivity of dc. In the second step, the NNE defined is used to mine the potential structure of data, which is useful to divide the dataset into several relatively more compact local groups called NNEGs. And the last step groups all NNEGs into the goal number of clusters using the proposed formula of the

closeness degree of local groups. And the application of the second and third steps not only overcomes the issue of remote assignment of the prominent density peaks but also removes the step of center selection in the original DPC. The effectiveness of the method proposed was verified on several datasets. The results show that our approach is more effective against the related improvement algorithms of DPC. In future work, we shall contribute to developing the concept of NNE to find a more suitable method for secondary-adjacent samples, instead of the given and fixated parameter $2K_i$ in equation (7). Fuzzy theory is a proper technique to mine relatively adjacent samples, in which NNE is used to construct the membership function of closeness, and then deduce the functions of secondary-adjacent samples and remote samples.

Data Availability

All datasets in this paper are available in UCI.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (61972056, 61772454, 61402053, and 61981340416), Hunan Provincial Natural Science Foundation of China (2020JJ4623), Scientific Research Fund of Hunan Provincial Education Department (17A007, 19C0028, and 19B005), Changsha Science and Technology Planning (KQ1703018, KQ1706064, KQ1703018-01, and KQ1703018-04), Junior Faculty Development Program Project of Changsha University of Science and Technology (2019QJCZ011), "Double First-class" International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology (2019IC34), Practical Innovation and Entrepreneurship Ability Improvement Plan for Professional Degree Postgraduate of Changsha University of Science and Technology (SJCX202072), Postgraduate Training Innovation Base Construction Project of Hunan Province (2019-248-51 and 2020-172-48), and Beidou Micro Project of Hunan Provincial Education Department (XJT(2020) No.149).

References

[1] A. Saxena, M. Prasad, A. Gupta et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[2] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier," *Cluster Computing*, vol. 22, no. 10, pp. 7665–7675, 2019.

[3] L. Sun, C. Ma, Y. Chen et al., "Low rank component induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, p. 3829, 2020.

[4] Y. Chen, J. Tao, Q. Zhang et al., "Saliency detection via improved hierarchical principle component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8822777, 12 pages, 2020.

[5] W. Lu, X. Zhang, H. Lu, and F. Li, "Deep hierarchical encoding model for sentence semantic matching," *Journal of Visual Communication and Image Representation*, vol. 71, Article ID 102794, 2020.

[6] Y. Chen, J. Wang, X. Chen et al., "Single-image super-resolution algorithm based on structural self-similarity and deformation block features," *IEEE Access*, vol. 7, pp. 58791–58801, 2019.

[7] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[8] Y. Chen, J. Wang, X. Chen, A. K. Sangaiah, K. Yang, and Z. Cao, "Image super-resolution algorithm based on dual-channel convolutional neural networks," *Applied Sciences*, vol. 9, no. 11, p. 2316, 2019.

[9] F. Yu, L. Liu, H. Shen et al., "Dynamic analysis, circuit design and synchronization of a novel 6D memristive four-wing hyperchaotic system with multiple coexisting attractors," *Complexity*, vol. 2020, Article ID 5904607, 17 pages, 2020.

[10] Y. Chen, L. Liu, J. Tao et al., "The improved image inpainting algorithm via encoder and similarity constraint," *The Visual Computer*, vol. 2020, 2020.

[11] L. Sun, F. Wu, T. Zhan, W. Liu, J. Wang, and B. Jeon, "Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1174–1188, 2020.

[12] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, and K. Yang, "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855–4867, 2019.

[13] Y. Zhang, W. Lu, W. Ou et al., "Chinese medical question answer selection via hybrid models based on CNN and GRU," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 14751–14776, 2020.

[14] Y. Chen, J. Xiong, W. Xu, and J. Zuo, "A novel online incremental and decremental learning algorithm based on variable support vector machine," *Cluster Computing*, vol. 22, no. 8, pp. 7435–7445, 2019.

[15] J. Wang, J. Qin, J. Qin, X. Xiang, Y. Tan, and N. Pan, "CAPTCHA recognition based on deep convolutional neural network," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851–5861, 2019.

[16] Y. Chen, J. Tao, L. Liu et al., "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2020, 2020.

[17] F. Yu, L. Liu, H. Shen et al., "Multistability analysis, coexisting multiple attractors and FPGA implementation of Yu-Wang four-wing chaotic system," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7530976, 16 pages, 2020.

[18] Y. Chen, J. Wang, S. Liu et al., "Multiscale fast correlation filtering tracking algorithm based on a feature fusion model," *Concurrency and Computation: Practice and Experience*, vol. 2019, 2019.

[19] A. K. Jain, "Data clustering: 50 years beyond *K*-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[20] D. Lam and D. C. Wunsch, *Academic Press Library in Signal Processing*Elsevier Press, Waltham, MA, USA, 2014.
[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, Berkeley, CA, USA, January 1967.
[22] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
[23] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
[24] G. McLachlan and D. Peel, "Finite mixture models," in *Encyclopedia of Autism Spectrum Disorders*p. 1296, 1st edition, Springer Press, Manhattan, NY, USA, 2013.
[25] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.
[26] J. Zhong, W. T. Peter, and Y. Wei, "An intelligent and improved density and distance-based clustering approach for industrial survey data classification," *Expert Systems with Applications*, vol. 68, pp. 21–28, 2017.
[27] S. Guha, R. Rastogi, and S. Kyuseok, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management Of Data ACM*, pp. 73–84, Seattle, WA, USA, June 1998.
[28] S. Guha, R. Rastogi, and K. Shim, "Rock: a robust clustering algorithm for categorical attributes," in *Proceedings of the IEEE Conference on Data Engineering*, pp. 512–521, Sydney, Australia, March 1999.
[29] W. Wang, J. Yang, and R. Muntz, "Sting: a statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 186–195, Athens, Greece, August 1997.
[30] M. Ester, H. P. Kriegel, J. Sander et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Portland, Oregon, August 1996.
[31] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *Proceedings of the ACM Sigmod Record*, pp. 49–60, Philadelphia, PA, USA, 1999.
[32] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[33] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted *k*-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.
[34] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognition*, vol. 107, Article ID 107449, 2020.
[35] L. Ni, W. Luo, W. Zhu, and W. Liu, "Clustering by finding prominent peaks in density space," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 727–739, 2019.
[36] Q. Zhu, J. Feng, and J. Huang, "Natural neighbor: a self-adaptive neighborhood method without parameter *K*," *Pattern Recognition Letters*, vol. 80, pp. 30–36, 2016.
[37] A. Lotfi, S. A. Seyedi, and P. Moradi, "An improved density peaks method for data clustering," in *Proceedings of the 6th International Conference on Computer and Knowledge Engineering*, pp. 263–268, Mashhad, Iran, October 2016.
[38] P. Guo, X. Wang, Y. Wang et al., "Research on automatic determining clustering centers algorithm based on linear regression analysis," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing*, pp. 1016–1023, Chengdu, China, August 2017.
[39] J. Ding, X. He, J. Yuan, and B. Jiang, "Automatic clustering based on density peak detection using generalized extreme value distribution," *Soft Computing*, vol. 22, no. 9, pp. 2777–2796, 2018.
[40] J. Jiang, Y. Chen, X. Meng, L. Wang, and K. Li, "A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process," *Physica A: Statistical Mechanics and Its Applications*, vol. 523, pp. 702–713, 2019.
[41] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[42] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowledge and Information Systems*, vol. 19, no. 3, pp. 361–394, 2009.
[43] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.
[44] P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: cluster level similarity measure," *Pattern Recognition*, vol. 47, no. 9, pp. 3034–3045, 2014.

Hindawi
Complexity
Volume 2020, Article ID 8854472, 12 pages
https://doi.org/10.1155/2020/8854472

Research Article

Multiphase SVPWM Strategy Analysis and Implementation of Seven-Phase Permanent Magnet Synchronous Motor

Mingli Lu,[1] **Dong Zhang**,[1] **Benlian Xu**,[2] **Haodong Yang**,[1] **and Yi Xin**[2]

[1]*School of Electrical & Automatic Engineering, Changshu Institute of Technology, Changshu 215500, China*
[2]*School of Mechanical Engineering, Changshu Institute of Technology, Changshu 215500, China*

Correspondence should be addressed to Benlian Xu; xu_benlian@cslg.edu.cn

Received 19 August 2020; Revised 19 September 2020; Accepted 8 October 2020; Published 21 October 2020

Academic Editor: Rui Wang

The multiphase motor drive systems have become a focus in many application areas such as ship electric propulsion, urban mass transit, aerospace, and weapon equipment, as they are characterized by high power density, low torque pulsation as small torque ripple, large output power, strong fault tolerance, and high reliability. However, with the increase of the phase number of the motor, the current harmonic component increases correspondingly, which leads to the decrease of the control performance compared with the three-phase system. In order to overcome this challenge, implementation method of driving control technology for seven-phase permanent magnet synchronous motor (PMSM) based on SVPWM algorithm is discussed thoroughly in this paper. Simulink and experiments have been developed to check its practical feasibility. The results show that the near-six vector SVPWM algorithm (NSV-SVPWM) achieves better performance than other methods.

1. Introduction

Multiphase PMSM is widely used in special occasions such as high current and high voltage, and it works in the case of low voltage and high power drive without any restriction of current or voltage. Compared with three-phase permanent magnet motor, PMSM has many advantages, such as simple structure, small volume, high power output density, high efficiency, low torque ripple, and low vibration noise, and it has been widely used in the field of micro electric vehicle [1–5]. As the number of motor phases increases, the pulse frequency of the output torque of the multiphase motor drive system increases, and the amplitude of the mechanical torque ripple decreases; thus, the static and dynamic performance of the multiphase motor drive system can be improved, and the reliability of the whole drive system is improved. For the dimension of the multiphase motor drive system, the three-dimensional system of the traditional three-phase motor is transformed into a multidimensional system. The increase of the dimension of the motor makes the control strategy of the multiphase motor more abundant, thus making the more complex motor control algorithm more easily realized.

In view of the above advantages of multiphase PMSM, the seven-phase PMSM is considered. However, due to the increase of the phases of motors, the current harmonic contents of multiphase motor system are larger, which leads to the decrease of the control performance compared with the three-phase system [6, 7]. The inverter nonlinearity is the key source of current harmonics. To solve this problem, domestic and foreign experts and scholars have conducted a lot of research [6, 8–10], such as starting from the design improvement of motor, changing the distribution of motor winding and distribution of groove to reduce the harmonic current, and starting from the angle of control strategy, current decoupling to different harmonic subspaces to reduce the harmonic content [6].

With regard to the PWM of seven-phase system, the SVPWM theory can be still employed to represent the behavior of seven-phase systems as a natural extension of the traditional three-phase SVPWM transformation [11, 12]. In this paper, the space vector modulation (SVM) has been extended to a seven-phase voltage source inverter, considering reference space vectors in all the three d-q planes [13]. Especially, the proposed SVM strategy univocally selects the

inverter switch configurations among the 128 possibility by privileging the space vector on the first subspace plane, $d1$-$q1$, the one responsible of balanced sinusoidal output voltage waveforms [14]. Then, vector analysis of the other two planes that produce the harmonic current is made to find out whether the vector that generates harmonics can be counteracted as much as possible. So, the implementation method of driving control algorithm based on SVPWM for low-order current harmonics suppression is discussed in this paper.

The goal of this paper is design a framework that addresses the above challenges. First, the mathematical model of the static coordinate system of the seven-phase PMSM is analyzed in detail. Then, a novel synchronous rotating coordinate transformation matrix was presented, and a mathematical model of the rotating coordinate system is obtained by matrix converter. And then, the 3rd harmonic subspace and the 5th harmonic subspace are derived based on the spatial voltage vector distribution of seven-phase bridge inverter. After analyzing the performance of these methods, namely, the near-four vectors SVPWM (NFV-SVPWM) algorithm and NSV-SVPWM algorithm, NSV-SVPWM algorithm is carried out. Finally, a driving control platform based on STM32F407 primary control chip is designed for seven-phase PMSM.

Preliminary results have been announced in the conference paper [15]. A more complete framework of control technology of seven-phase PMSM for low-order current harmonics suppression is proposed. We also describe in detail the experiments analysis. The rest of the paper is organized as follows. Section 2 gives mathematic model of seven-phase PMSM. The seven-phase SVPWM algorithm is described in Section 3. Simulations and experiments analysis of PMSM and the detailed discussion are described in Section 4. Finally, a summary is provided in Section 5.

2. Mathematic Model of Seven-Phase PMSM

The investigated motor is a hidden pole type seven-phase PMSM. Before studying the driving control system of the seven-phase PMSM, the mathematical model of the motor is established at first. Therefore, for the convenience of analysis, the following assumptions are made for the seven-phase PMSM studied:

(1) Neglect the influence of eddy current and hysteresis in iron core

(2) The magnetic circuit of the motor is linear, and the saturation effect will not occur

(3) The stator surface of the motor is smooth

(4) Neglect influence of motor temperature on motor winding

(5) The structure of the motor is implicit, and the reactance is equal

(6) The seven-phase winding is "Y" type connection, and the spatial distribution is completely symmetrical

In the stationary coordinate system, the flux equation, the stator voltage equation, the electromagnetic torque equation, and the mechanical motion equation are important mathematical model of the seven-phase PMSM.

In natural coordinate, the voltage equation for seven-phase PMSM is given:

$$U_v = R_v I_v + g\psi_v, \tag{1}$$

where $U_v = \left[u_a,\ u_b,\ u_c,\ u_d,\ u_e,\ u_f,\ u_d \right]^T$ is the stator voltage matrix, R_v represents the stator resistance matrix, $I_v = \left[i_a,\ i_b,\ i_c,\ i_d,\ i_e,\ i_f,\ i_d \right]^T$ is denoted as the stator current matrix, and $g = \mathrm{d}/\mathrm{d}t$. $\psi_v = \left[\psi_1,\ \psi_2,\ \psi_3,\ \psi_4,\ \psi_5,\ \psi_6,\ \psi_7 \right]^T$ is interpreted as the stator flux matrix.

The flux equation for seven-phase PMSM is given:

$$\psi_v = L_v I_v + \psi_n, \tag{2}$$

where L_v is the stator inductance matrix, and Ψ_n represents the flux linkage of permanent magnet. It can be defined as

$$\psi_n = \psi \begin{bmatrix} \cos\theta \\ \cos(\theta - \alpha) \\ \cos(\theta - 2\alpha) \\ \cos(\theta - 3\alpha) \\ \cos(\theta - 4\alpha) \\ \cos(\theta - 5\alpha) \\ \cos(\theta - 6\alpha) \end{bmatrix}, \tag{3}$$

where ψ, θ are the flux linkage amplitude of permanent magnet and the rotor position angle, respectively. $\alpha = 2\pi/7$ is the angle between adjacent two-phase winding axes.

The torque equation for seven-phase PMSM is given:

$$\begin{cases} \mathrm{M}_{\mathrm{eo}} = \dfrac{1}{2} I_v^T L_v I_s + I_v^T \psi_n, \\ \\ T_{\mathrm{e}} = a \dfrac{\mathrm{dM}_{\mathrm{eo}}}{\mathrm{d}\theta}, \end{cases} \tag{4}$$

where T_{e} is motor electromagnetic torque, M_{eo} is denoted as motor magnetic energy, and a is the polar logarithm of motor.

For the hidden pole rotor of motor, L_v and θ are independent of each other, then the above function can be rewritten as

$$T_{\mathrm{e}} = a \frac{\mathrm{dM}_{\mathrm{eo}}}{\mathrm{d}\theta} = a \left[I_v \frac{\mathrm{d}\psi_n}{\mathrm{d}\theta} \right]. \tag{5}$$

The mechanical motion equation for seven-phase PMSM is

$$\frac{J}{a} \frac{\mathrm{d}\omega_r}{\mathrm{d}t} = T_{\mathrm{e}} - T_{\mathrm{l}}, \tag{6}$$

where T_l is denoted as motor load torque, and J is moment of inertia of mechanical and electrical system. Seven-phase

PMSM is a nonlinear system. The methods [16, 17] can provide a reference for some parameters estimated and model constructed.

According to equations (2)–(6) of electromagnetic torque equation, it can be seen that the straight axis and quadrature axis currents of stator of seven-phase PSMS are coupled with each other, which brings inconvenience to the subsequent research on vector control theory. Therefore, it is necessary to select appropriate spatial transformation and simplify the mathematical model of the seven-phase PMSM.

According to the principle of constant power, the transformation matrix from natural coordinate system to rotating coordinate system $(d_1 - q_1 - d_3 - q_3 - d_5 - q_5 - 0)$ based on the matrix transformation of Clark and Park is given by

$$\Pi(\theta) = \sqrt{\frac{2}{7}} \begin{bmatrix} \cos\theta & \cos(\theta-\alpha) & \dots & \cos(\theta+\alpha) \\ -\sin\theta & -\sin(\theta-\alpha) & \dots & -\sin(\theta+\alpha) \\ \cos 3\theta & \cos 3(\theta-\alpha) & \dots & \cos 3(\theta+\alpha) \\ -\sin 3\theta & -\sin 3(\theta-\alpha) & \dots & -\sin 3(\theta+\alpha) \\ \cos 5\theta & \cos 5(\theta-\alpha) & \dots & \cos 5(\theta+\alpha) \\ -\sin 5\theta & -\sin 5(\theta-\alpha) & \dots & -\sin 5(\theta+\alpha) \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{7}$$

If equation (7) is introduced into equations (2)–(6), through matrix transformation, the mathematical model of the motor in the rotating coordinate system can be obtained.

The voltage equation for seven-phase PMSM is given by

$$\begin{aligned} U_{\mathrm{dq}} &= \Pi(\theta)U_v = \Pi(\theta)R_v I_v + \Pi(\theta)g\Psi_v \\ &= \Pi(\theta)R_v\Pi(\theta)^{-1}\Pi(\theta)I_v + g\left[\Pi(\theta)\Psi_v\right] - g\Pi(\theta)\Pi(\theta)^{-1}\Pi(\theta)\Psi_v \\ &= R_{\mathrm{dq}}I_{\mathrm{dq}} + g\Psi_{\mathrm{dq}} + \Lambda\Psi_{\mathrm{dq}}, \end{aligned} \tag{8}$$

where $U_{\mathrm{dq}} = \Pi(\theta)U_v = \left[\, U_{d1} \;\; U_{q1} \;\; U_{d3} \;\; U_{q1} \;\; U_{d1} \;\; U_{q3} \;\; 0 \,\right]^T$

$$\begin{aligned} R_{\mathrm{dq}} &= \Pi(\theta)R_v\Pi(\theta)^{-1} = r_v I_{7\times 7}, \\ I_{\mathrm{dq}} &= \Pi(\theta)I_v = \left[\, i_{d1} \;\; i_{q1} \;\; i_{d3} \;\; i_{q1} \;\; i_{d1} \;\; i_{q3} \;\; 0 \,\right]^T, \\ \Psi_{\mathrm{dq}} &= \Pi(\theta)\Psi_v = \left[\, \Psi_{d1} \;\; \Psi_{q1} \;\; \Psi_{d3} \;\; \Psi_{q1} \;\; \Psi_{d1} \;\; \Psi_{q3} \;\; 0 \,\right]^T. \end{aligned} \tag{9}$$

In this way, other equations can be obtained. The flux equation for seven-phase PMSM is given by

$$\Psi_{\mathrm{dq}} = \Pi(\theta)\Psi_v = \Pi(\theta)\left(L_v I_v + \Psi_n\right). \tag{10}$$

For the hidden pole rotor of motor, the torque equation can be calculated as

$$\begin{aligned} T_{\mathrm{e}} &= \frac{\partial \mathrm{M}_{\mathrm{eo}}}{\partial\theta} = a\left[I_v\frac{\partial\Psi_n}{\partial\theta}\right] \\ &= \frac{7}{2}aI_{\mathrm{dq}}^T\Pi(\theta)\frac{\partial\Psi_n}{\partial\theta} \\ &= \frac{7}{2}\alpha\Psi i_{q1}, \end{aligned} \tag{11}$$

where i_{q1} is the component of stator current on the q axis.

3. Seven-Phase SVPWM Algorithm

3.1. Seven-Phase Bridge Inverter. The seven-phase motor drive systems are generally supplied by a seven-phase voltage source bridge inverter. Similar to three-phase bridge inverter, the topology of seven-phase bridge inverter is connected by 14 power switch devices, as shown in Figure 1.

In order to approach the circular magnetic field well, there should be as many regular polygon edges as possible; that is, more inverter switching state combinations should be generated. For seven-phase voltage inverter, the state variable S_i $(i = a, b, c, d, e, f, g)$ is introduced to represent the switching state of each bridge arm, and the turn-off and turn-on of the switch are represented by "0" and "1." Then, the seven-phase voltage type inverter can be combined into $27 = 128$ switching states [18, 19].

In a multiphase voltage source inverter (VSI), switching states determine load equivalent circuit configurations. Therefore, switching states can be divided into corresponding sets. In a seven-phase inverter, these basic equivalent circuit configurations can be categorized into three types, namely, $C16$ equivalent circuit, $C25$ equivalent circuit, and $C34$ equivalent circuit, as shown in Figure 2. Therefore, the switching states can be represented by three sets: $\{C16\}$, $\{C25\}$ and $\{C34\}$.

From the above, the space vector is obtained as

$$\begin{cases} U_{v1} = \dfrac{2}{7}U_{\mathrm{dc}}\left(S_a + \alpha S_b + \alpha^2 S_c + \alpha^3 S_d + \alpha^4 S_e + \alpha^5 S_f + \alpha^6 S_g\right), \\ U_{v3} = \dfrac{2}{7}U_{\mathrm{dc}}\left(S_a + \alpha^3 S_b + \alpha^6 S_c + \alpha^2 S_d + \alpha^5 S_e + \alpha S_f + \alpha^4 S_g\right), \\ U_{v5} = \dfrac{2}{7}U_{\mathrm{dc}}\left(S_a + \alpha^5 S_b + \alpha^3 S_c + \alpha S_d + \alpha^6 S_e + \alpha^4 S_f + \alpha^2 S_g\right). \end{cases} \tag{12}$$

Aiming to produce the desired fundamental (sinusoidal) component in the output phase voltage without low-order harmonics, SVPWM algorithm is used. Supposing that the DC bus voltage is considered as "1," Table 1 presents the magnitude relation of the voltage vector.

3.2. Current Harmonics Suppression Method. Current harmonics (3rd, 5th) caused by the inverter nonlinearity in phase currents are considered. To increase efficiency and

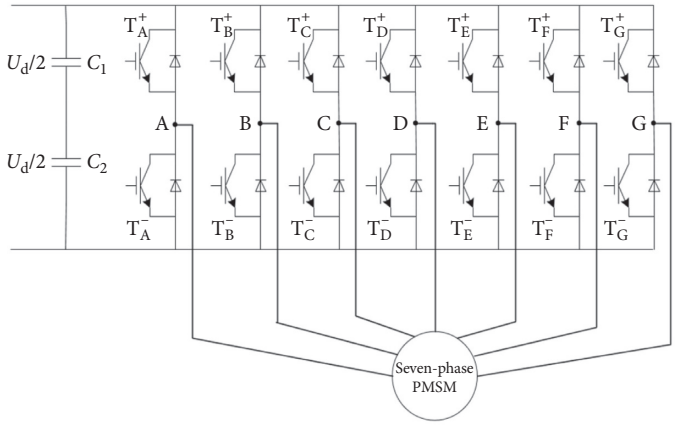


Figure 1: Main circuit of seven-phase bridge inverter.

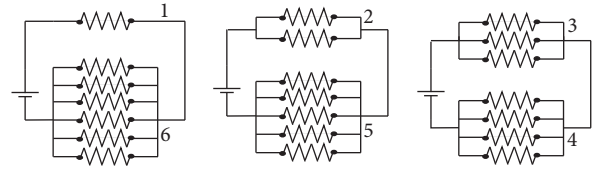


Figure 2: Switch combination equivalent circuit.

Table 1: Voltage vector amplitude relation table.

Vectors	U_a	U_b	U_c	U_d	U_e	U_f	U_g
Amplitude	$0.127U_{dc}$	$0.159U_{dc}$	$0.229U_{dc}$	$0.286U_{dc}$	$0.356U_{dc}$	$0.515U_{dc}$	$0.642U_{dc}$

performance of multiphase power converters, the current total harmonic should be eliminated, which highly relies on their control strategy. In this section, two control strategies of SVPWM algorithm are presented in detail.

3.2.1. The NFV-SVPWM Algorithm. Through analysis of the near-four vectors SVPWM (NTV-SVPWM) algorithm [15], it is found that the medium and small vector groups of the space voltage vector in the seven-phase motor system are not fully utilized to remove the harmonic component of the phase current. Therefore, we take into account the more abundant basic voltage vectors in the seven-phase inverter to reduce the harmonic component. Two near vectors, two large vectors, and zero vectors in the fundamental wave subspace are used to synthesize the reference voltage vector, that is, NFV-SVPWM algorithm [20], whose control strategy is the same as the three-phase SVPWM. The reference voltage in the sector is obtained by using the parallelogram rule synthesis. The maximum voltage vector and the middle vector correspond to the U_g vector group and the U_f vector group in Table 1, respectively.

The NFV-SVPWM algorithm of seven-phase motor system follows the following principles: The selected basic voltage vectors are given the voltage reference vector in the synthetic fundamental wave space, and the voltage vector synthesized in the 3rd harmonic subspace should be guaranteed to be zero. As shown in Figure 3, in the basic wave space, voltage vectors U_{71}, U_3 and U_{67}, U_{103} are in the same direction, while in the 3rd harmonic subspace, voltage vectors U_{71}, U_3 and U_{67}, U_{103} are opposite to each other in direction. In order to make the voltage vector of the 3rd harmonic subspace zero, the action time of voltage vectors U_{71}, U_3 and U_{67}, U_{103} is inversely proportional to the amplitude of the voltage vector in the 3rd harmonic subspace. In this way, the voltage vectors cancel each other without affecting the synthesis of voltage vectors in fundamental wave space.

For example, in the first sector, fundamental voltage vectors $U_3, U_{67}, U_{71}, U_{103}$ and U_0, U_{127} two zero vectors are

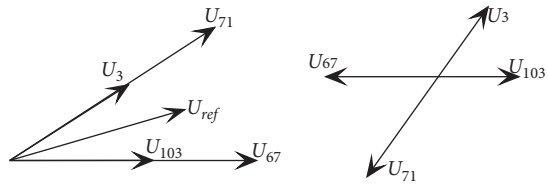


Figure 3: NFV-SVPWM vector synthesis. (a) Fundamental wave subspace. (b) The 3rd harmonic subspace.

selected to synthesize the reference vector U_{ref}. The time of action is defined as T_1, T_2, T_3, T_4 and T_0, respectively.

Similar to the NTV-SVPWM analysis method, the following relations can be obtained:

$$\begin{cases} T_1 = \dfrac{U_\alpha \sin[(m-1)(\pi/7)] - U_\beta \cos[(m-1)(\pi/7)]}{\left(1+1.2466^2\right)\left|U_M\right|\{\cos(m\pi/7)\sin[(m-1)(\pi/7)] - \sin(m(\pi/7))\cos[(m-1)(\pi/7)]\}} T_s, \\ T_2 = 1.2466 T_4, \\ T_3 = 1.2466 T_1, \\ T_4 = \dfrac{U_\alpha \sin(m(\pi/7)) - U_\beta \cos(m(\pi/7))}{\left(1+1.2466^2\right)\left|U_M\right|\{\cos(m(\pi/7))\sin[(m-1)(\pi/7)] - \sin(m(\pi/7))\cos[(m-1)(\pi/7)]\}} T_S. \end{cases} \tag{13}$$

To reduce the switching losses, the switching signal of the voltage vector in sector one is shown in Figure 4.

3.2.2. The NSV-SVPWM Algorithm. It is well known that 6 groups of voltage vector can synthesize the desired voltage vector. However, voltage vectors U_a, U_b, U_c and U_e with small amplitude are not continuously switched, which lead to desired voltage vector different direction. So, the voltage vector of the stator flux may offset. In NSV-SVPWM algorithm, the reference vector needs six voltage vectors to be synthesized in each sector of the fundamental wavelet subspace. The previous analysis shows that only three vectors U_d, U_f, and U_g of seven groups of voltage vectors in the fundamental subspace are used. Therefore, NSV-SVPWM algorithm is used to synthesize the reference voltage vector by a total of three groups of six voltage vectors: U_d, U_f, and U_g.

For example, in the first sector, select fundamental voltage vectors U_1, U_3, U_{67}, U_{71}, U_{103}, U_{111} and two zero vectors U_0, U_{127} to synthesize the reference vector U_{ref}. The time of action is defined as T_1, T_2 T_3, T_4, T_5, T_6 and T_0, respectively.

Aiming to cut down the harmonic content of phase current, the voltage vector synthesized in the $\alpha_3 - \beta_3$ and $\alpha_5 - \beta_5$ subspaces should be zero. As shown in Figure 5, it can be seen that voltage vectors U_1, U_{67} and U_{103} are in the same direction in $\alpha_1 - \beta_1$ subspace, while vector U_{67} is opposite to the corresponding two vectors in the $\alpha_3 - \beta_3$ and $\alpha_5 - \beta_5$ subspaces. To make the synthesized voltage vectors be zero in the subspaces $\alpha_3 - \beta_3$ and $\alpha_5 - \beta_5$, the reverse vector can be counteracted by the corresponding two other vectors. Thus, the resultant voltage of each direction is zero, and the current harmonic is suppressed.

In the $\alpha_1 - \beta_1$ fundamental subspace, the following equations came up

$$\begin{cases} U_{\alpha 1} = \dfrac{T_1}{T_s}\left|U_1\right| + \dfrac{T_2}{T_s}\left|U_3\right|\cos\dfrac{\pi}{7} + \dfrac{T_3}{T_s}\left|U_{67}\right| + \dfrac{T_4}{T_s}\left|U_{71}\right|\cos\dfrac{\pi}{7} + \dfrac{T_5}{T_s}\left|U_{103}\right| + \dfrac{T_6}{T_s}\left|U_{111}\right|\cos\dfrac{\pi}{7}, \\ U_{\beta 1} = \dfrac{T_2}{T_s}\left|U_3\right|\sin\dfrac{\pi}{7} + \dfrac{T_4}{T_s}\left|U_{71}\right|\sin\dfrac{\pi}{7} + \dfrac{T_6}{T_s}\left|U_{111}\right|\sin\dfrac{\pi}{7}. \end{cases} \tag{14}$$

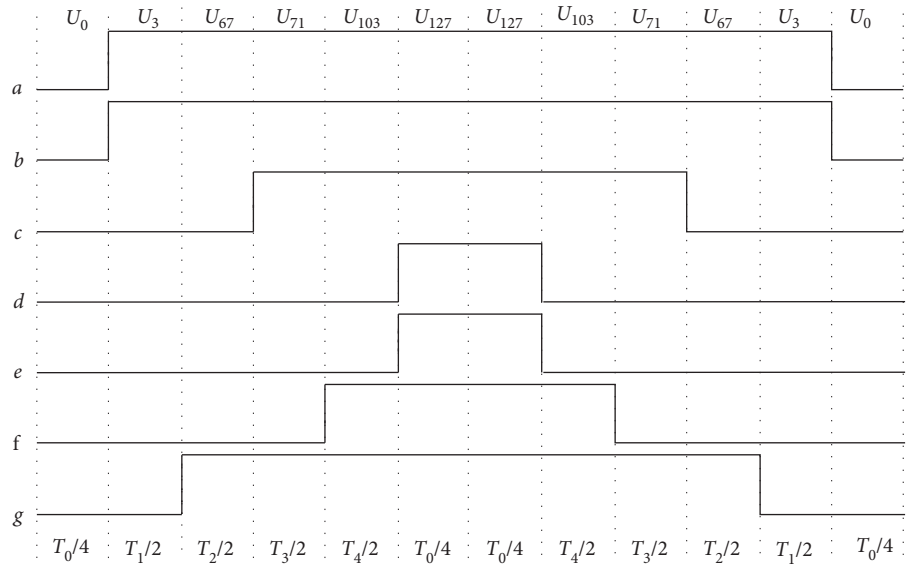


Figure 4: Switching signal of NFV-SVPWM algorithm in sector one.

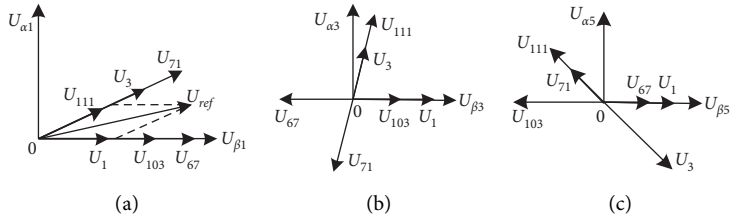


Figure 5: NSV-SVPWM vector synthesis: (a) Fundamental wave subspace. (b) The 3rd harmonic subspace. (c) The 5th harmonic subspace.

In the 3rd harmonic subspaces, the following equations are obtained:

$$\begin{cases} \dfrac{T_1}{T_s}|U_1| + \dfrac{T_5}{T_s}|U_{103}| - \dfrac{T_3}{T_s}|U_{67}| = 0, \\ \dfrac{T_2}{T_s}|U_3| + \dfrac{T_6}{T_s}|U_{111}| - \dfrac{T_4}{T_s}|U_{71}| = 0. \end{cases} \tag{15}$$

In the same way, the following equations came up in the 5th harmonic subspace

$$\begin{cases} \dfrac{T_1}{T_s}|U_1| + \dfrac{T_3}{T_s}|U_{67}| - \dfrac{T_5}{T_s}|U_{103}| = 0, \\ \dfrac{T_6}{T_s}|U_{111}| + \dfrac{T_4}{T_s}|U_{71}| - \dfrac{T_2}{T_s}|U_3| = 0. \end{cases} \tag{16}$$

According to the vector magnitude of Table 1, the substitution formulae (14)–(16), a set of equations with six variables and solving the equations can be expressed as

$$\begin{cases} T_1 = \dfrac{U_{\alpha1}\sin(\pi/7) - U_{\beta1}\cos(\pi/7)}{2.6638U_{\mathrm{dc}}\sin(\pi/7)}T_s, \\ T_2 = \dfrac{U_{\beta1}}{1.4773U_{\mathrm{dc}}\sin(\pi/7)}T_s, \\ T_3 = 2.2524T_1, \\ T_4 = 2.2524T_2, \\ T_5 = 1.8093T_1, \\ T_6 = 1.8093T_2. \end{cases} \tag{17}$$

In this way, the voltage equations are listed for each sector based on the fundamental subspace $(\alpha_1 - \beta_1)$, 3rd harmonic subspace $(\alpha_3 - \beta_3)$ and 5th harmonic subspace $(\alpha_5 - \beta_5)$, respectively. These equations are simplified to obtain a new six-element first-order functions within each sector. For example, in sector m $(1 \le m \le 14)$, the following relation holds

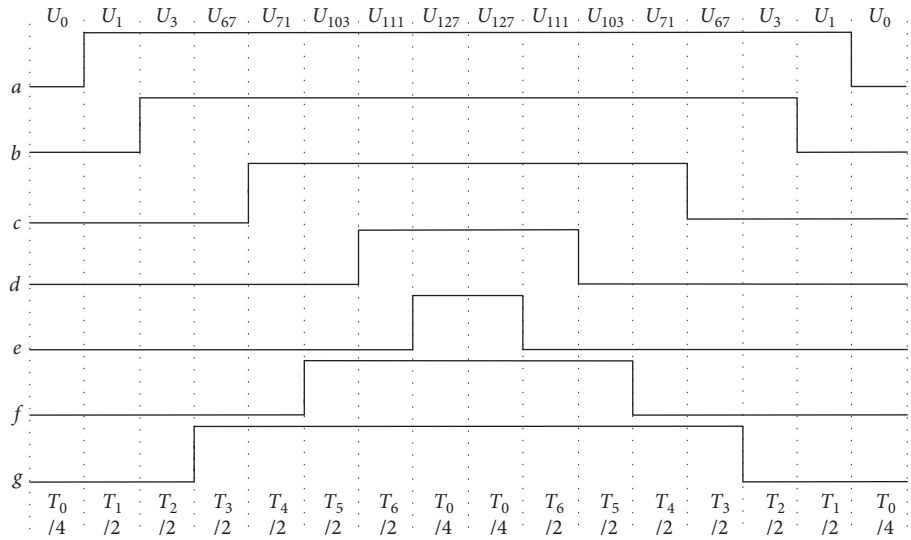


Figure 6: Switching signal of NSV-SVPWM algorithm in sector one.

$$\begin{cases} T_1 = \dfrac{U_{\beta 1}\cos[(m-1)(\pi/7)] - U_{\alpha 1}\sin[(m-1)(\pi/7)]}{1.1558U_{\mathrm{dc}}}T_s, \\ T_2 = \dfrac{U_{\alpha 1}\sin[(m-1)(\pi/7)] - U_{\beta 1}\cos[(m-1)(\pi/7)]}{0.6410U_{\mathrm{dc}}}T_s, \\ T_3 = 2.2524T_1, \\ T_4 = 2.2524T_2, \\ T_5 = 1.8093T_1, \\ T_6 = 1.8093T_2. \end{cases} \tag{18}$$

In order to reduce the switching losses, the voltage vectors for each sector should be reasonably arranged. Sector one ($m = 11$) is taken as an example for analysis. The order of action of the space voltage vector is arranged as follows: $U_0 \longrightarrow U_1 \longrightarrow U_0 \longrightarrow U_1 \longrightarrow U_3 \longrightarrow U_{67} \longrightarrow U_{71} \longrightarrow U_{103} \longrightarrow U_{111} \longrightarrow U_{71} \longrightarrow U_{67} \longrightarrow U_3 \longrightarrow U_1 \longrightarrow U_0$. The switching signal of the voltage vector in sector one is shown in Figure 6.

$$\begin{aligned} T_s &= T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_0, \\ U_{\mathrm{ref}}T_s &= T_1U_1 + T_2U_3 + T_3U_{67} + T_4U_{71} + T_5U_{103} + T_6U_{111} + T_0. \end{aligned} \tag{19}$$

4. Simulations and Experiments Analysis

4.1. Simulations and Analysis. The proposed algorithm is implemented in Matlab/Simulink platform. The performance of the three control algorithms (NTV-SVPWM, NFV-SVPWM, and NSV-SVPWM) mentioned above section is analyzed. The parameters of the seven-phase PMSM are consistent with the mathematical model. The motor has a load of 1 N.m, the switching frequency is 20 kHz, and the DC bus voltage is 72V. The stator resistance is 1.78Ω, and d-axis inductance L_d and q-axis inductance L_q are 4.5×10^{-3} H and 4.5×10^{-3} H, respectively. The permanent magnet flux linkage ψ_f is 0.175 Wb. The moment of inertia J is $0.4 \times 10^{-3}\,\mathrm{kg \cdot m^2}$.

Simulation results of NTV-SVPWM are shown in Figure 7. In Figures 7(a) and 7(b), x-coordinate is times (s) and y-coordinate is phase current (A). It can be seen that the phase current is less sinusoidal, and the FFT harmonic analysis shows that the 3rd and 5th harmonic contents are as high as 36.65% and 17.22%, respectively, and the total harmonic THD is 40.61%. The results show that the seven-phase system extended from traditional three-phase SVPWM algorithm has the problem of harmonic phase current.

Figure 8 presents the simulation results of NFV-SVPWM. Although the sinusoidal component of the phase current is slightly higher than the NTV-SVPWM algorithm, the sinusoidal phase current is still very poor as shown in Figure 8(b). Through the FFT harmonic analysis, it can be seen that the 3rd harmonic content is much less than that of the NTV-SVPWM algorithm, only 5.31%, while the 5th harmonic content is still large, 29.13%, and the overall THD is 16.01%. Though the NFV-SVPWM algorithm controls the influence of the harmonic subspace, only the 3rd harmonic subspace is considered, and the interference of the 5th harmonic subspace is not considered, which leads to the greater content of the 5th harmonic in the phase current. In Figures 8(a) and 8(b), x-coordinate is times (s) and y-coordinate is phase current (A).

Figure 9 provides the simulation results of NSV-SVPWM. Using the same motor parameters, the simulation results show that the phase current of NSV-SVPWM appears with tiny distortion, as shown in Figure 9(b). At the same time, the phase voltage waveform is saddle waveform, as shown in Figure 9(c), where x-coordinate is times (s) and

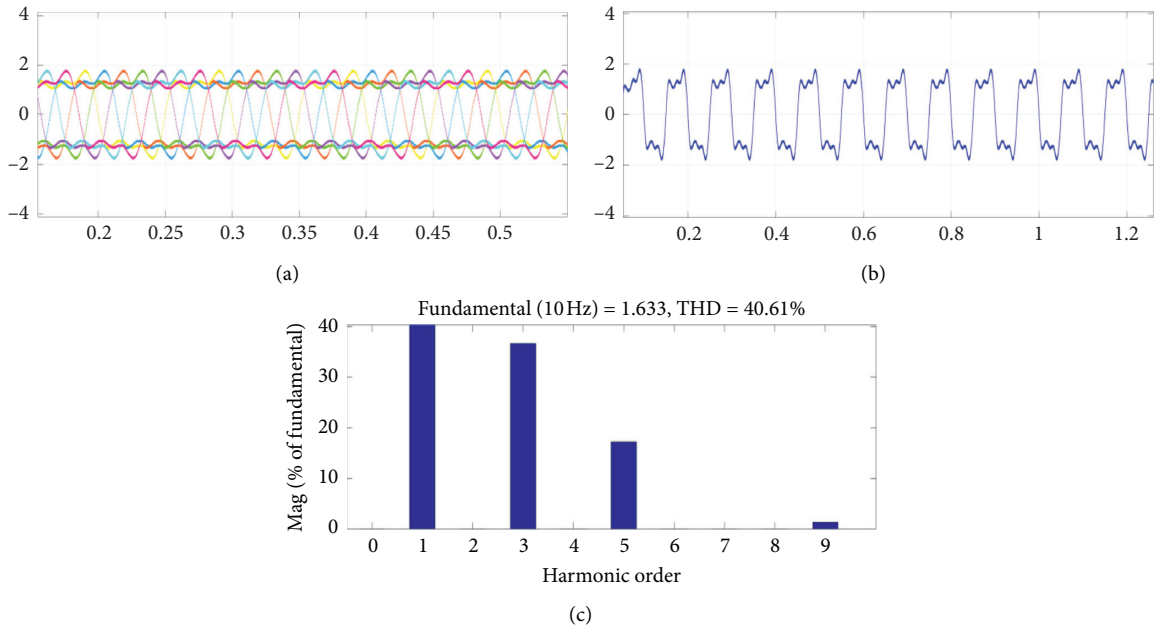


Figure 7: Simulation results of NTV-SVPWM, (a) seven-phase current waveform, (b) a phase current waveform, (c) FFT analysis.

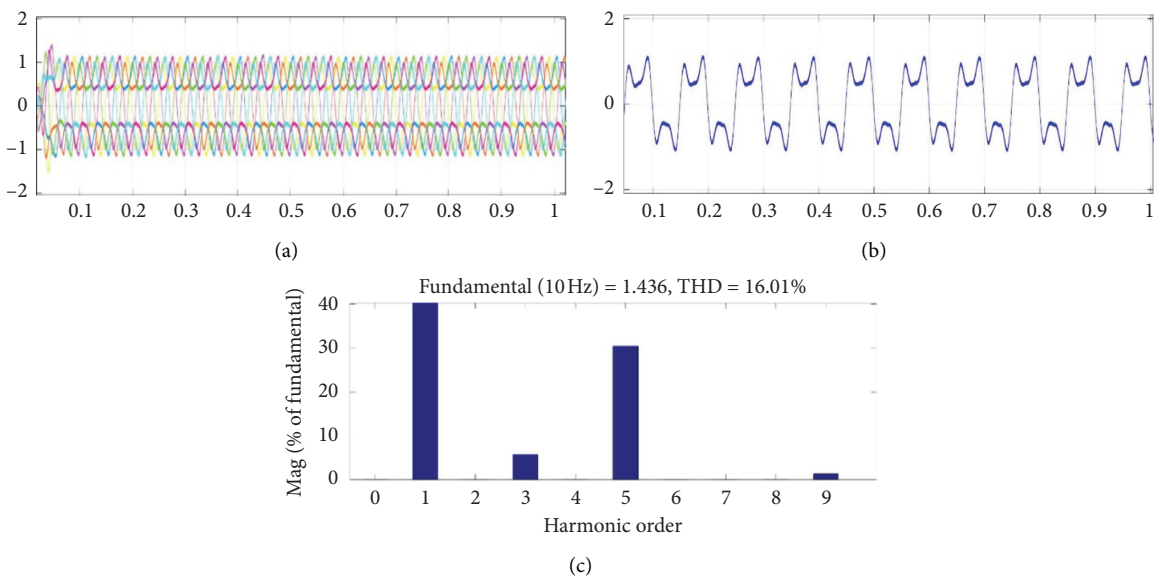


Figure 8: Simulation results of NFV-SVPWM, (a) seven-phase current waveform, (b) a phase current waveform, (c) FFT analysis.

y-coordinate is phase voltage (V). From the FFT harmonic analysis, it can be seen that the 3rd and 5th harmonic currents are much lower than those of the NTV-SVPWM algorithm, as shown in Figure 9(d). The harmonic contents of 3rd and 5th currents are only 4.29% and 1.76%, and the total THD is 3.51%.

According to the simulation results of above three algorithms, we can see that NSV-SVPWM achieves much better performance than the other methods.

4.2. Experiments and Analysis. A prototype of the proposed algorithm was then built in the lab and tested. In order to ensure that the driving system has good real-time, economic and good motor control characteristics, the STM32F407VET6 chip of the ARM Cotex-M4 kernel as the main control chip of the drive system has been used. Combined with the rated parameters of seven-phase PMSM, a motor experimental platform with a rated voltage of 72 V is built, and its physical map is shown in Figure 10.

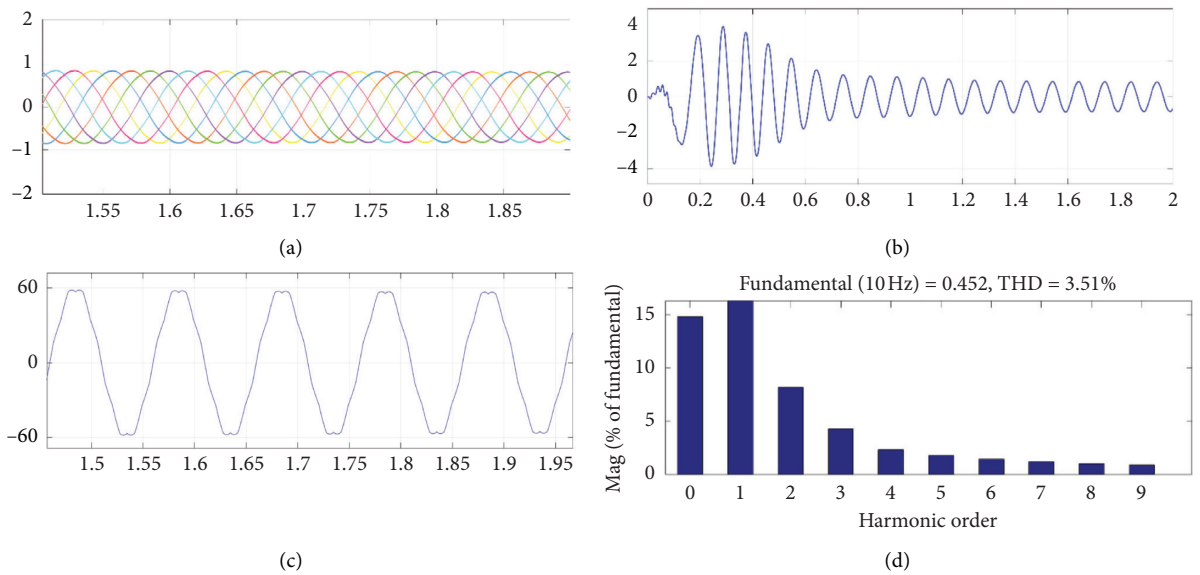


Figure 9: Simulation results of NSV-SVPWM, (a) seven-phase current waveform, (b) phase current waveform, (c) phase voltage waveform, (d) FFT analysis.

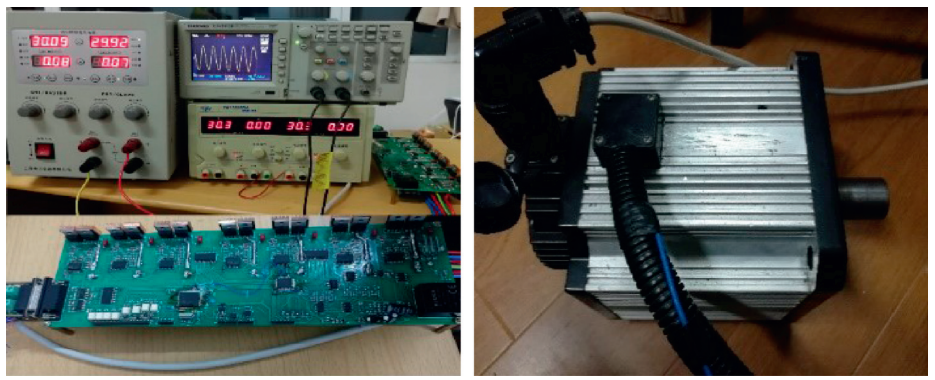
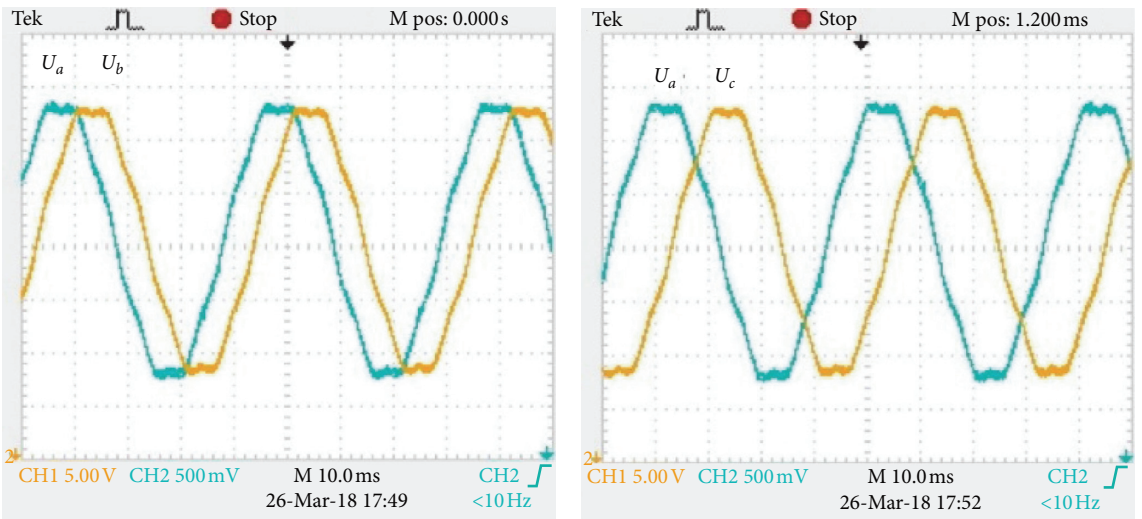


Figure 10: Seven-phase PMSM experiment platform.



(a) (b)

Figure 11: Continued.

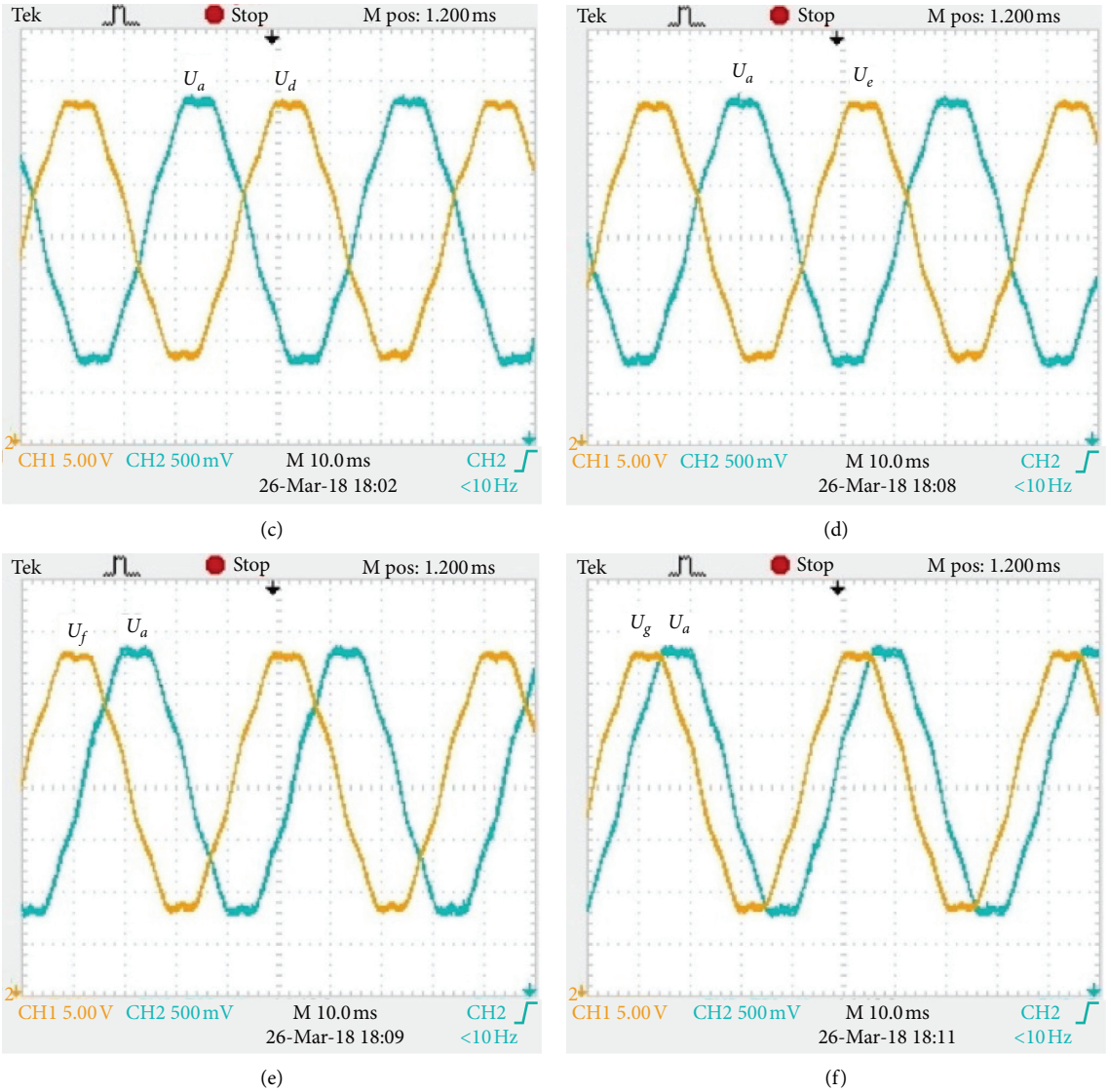


Figure 11: Phase voltage experimental waveform of NSV-SVPWM. (a) a phase and b phase voltage (b) a phase and c phase voltage, (c) a phase and d phase voltage, (d) a phase and e phase voltage, (e) a phase and f phase voltage, (f) a phase and g phase voltage.

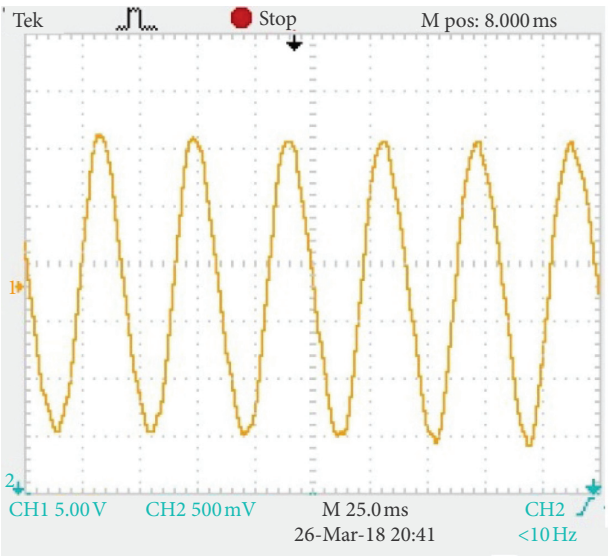


Figure 12: Phase current experimental waveform of NSV-SVPWM.

The phase voltage waveform of the NSV-SVPWM control mode is shown in Figure 11. It is not difficult to see that the phase voltage waveform is saddle wave, in which the phase difference between a phase voltage U_a and b phase voltage U_b is $2\pi/7$. The difference between a phase voltage U_a and c phase voltage U_c is $4\pi/7$, as shown in Figure 11(b). In a similar fashion, a phase voltage U_a lags g phase voltage U_g $2\pi/7$, as shown in Figure 11(f). Therefore, on the basis of the experimental results, the phase voltage waveform is consistent with the simulation results.

Figure 12 presents the phase current experimental waveform i_a. The sinusoidal component of phase current is very high and without current harmonics. It proves the feasibility of the NSV-SVPWM algorithm in the seven-phase PMSM drive control system. The experiment results are in agreement with the theoretical analysis and simulation results.

5. Conclusion

Multiphase PMSM is needed to produce the sinusoidal component in the output phase voltage, without low-order harmonics. In this paper, the rotating coordinate system of the seven-phase PMSM is obtained by matrix converter. To eliminate low-order harmonics, NTV-SVPWM algorithm, NFV-SVPWM algorithm, and NSV-SVPWM algorithm are discussed thoroughly based on simulations and experiments analysis. According to the simulation and experimental results, it can be concluded that the NSV-SVPWM algorithm not only suppresses the current high order harmonics, but also has excellent control performance.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Nos. 61673075 and 61876024), and Partly by the Project of Talent Peak of Six Industries (2017-DZXX-001), 333 Project of Jiangsu Province (No. BRA2019284), and The Science and Technology Development Plan Project of Chang Shu (CR0201711).

References

[1] S. Ye, "Design and performance analysis of an iterative flux sliding-mode observer for the sensorless control of PMSM drives," *ISA Transactions*, vol. 94, pp. 255–264, 2019.

[2] X. Kuang, H. Guo, J. Xu, and T. Zhou, "Research on a six-phase permanent magnet synchronous motor system at dual-redundant and fault tolerant modes in aviation application," *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1548–1560, 2017.

[3] A. Hosseyni, R. Trabelsi, M. F. Mimouni, A. Iqbal, and R. Alammari, "Sensorless sliding mode observer for a five-phase permanent magnet synchronous motor drive," *ISA Transactions*, vol. 58, pp. 462–473, 2015.

[4] H. Xu, S. Xue, and C. Fang, "Multiphase permanent magnet synchronous motor harmonic control base on carrier-based PWM technology," in *Proceedings of the IEEE Conference and Expo Transportation Electrification Asia-Pacific*, pp. 1–7, Beijing, China, September 2014.

[5] S. Xue, H. Xu, and C. Fang, "Multiphase permanent magnet synchronous motor harmonic plane control based on carrier technology," *Journal of Harbin Institute of Technology*, vol. 46, no. 4, pp. 122–128, 2014.

[6] L. Yuan, M. Chen, J. Shen, and F. Xiao, "Current harmonics elimination control method for six-phase PM synchronous motor drives," *ISA Transactions*, vol. 59, pp. 433–449, 2015.

[7] S. Xue and X. Wen, "Simulation analysis of a novel multiphase SVPWM strategy," in *Proceedings of the 2006 2nd International Conference on Power Electronics and Drives Systems*, pp. 756–760, Hong Kong, China, November 2006.

[8] O. Lopez, J. Alvarez, J. Malvar et al., "Space-vector PWM with common-mode voltage elimination for multiphase drives," *IEEE Transactions on Power Electronics*, vol. 31, no. 12, pp. 8151–8161, 2016.

[9] S. M. Dabour, S. M. Allam, and M. Essam, "Rashad space vector PWM technique for a novel three-to-seven phase matrix converter," in *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4949–4954, Vienna, Austria, November 2013.

[10] O. Dordevic, E. Levi, and M. Jones, "A vector space decomposition based space vector PWM algorithm for a three-level seven-phase voltage source inverter," *IEEE Transactions on Power Electronics*, vol. 28, no. 2, pp. 637–649, 2013.

[11] F. Yu, X. Zhang, and H. Li, "The space vector PWM control research of a multiphase permanent magnet synchronous motor for electrical propulsion," *International Conference on Electrical Machines and Systems*, vol. 2, pp. 604–607, 2003.

[12] G. Grandi, G. Serra, and A. Tani, "Space vector modulation of a seven-phase voltage source inverter," in *Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pp. 1149–1156, Taormina, Italy, June 2006.

[13] J. Pradeep and R. Devanathan, "Comparative analysis and simulation of PWM and SVPWM inverter fed permanent magnet synchronous motor," in *Proceedings of the International Conference on Emerging Trends in Electrical Engineering and Energy Management*, pp. 299–305, Roorkee, India, Feburary 2013.

[14] D. Casadei, D. Dujic, E. Levi, G. Serra, A. Tani, and L. Zarri, "General modulation strategy for seven-phase inverters with independent control of multiple voltage space vectors," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 5, pp. 1921–1932, 2008.

[15] D. Zhang, B. Xu, H. Yang, and P. Zhu, "Simulation analysis of SVPWM based on seven-phase permanent magnet synchronous motor," in *Proceedings of the International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 251–256, Ansan, South Korea, October 2017.

[16] W. Gong, Y. Wang, Z. Cai, and L. Wang, "Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1499–1513, 2020.

[17] S. Li, W. Gong, X. Yan et al., "Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization," *Energy Conversion and Management*, vol. 186, pp. 293–305, 2019.

[18] S. M. Dabour, "Space vector PWM technique to reduce common mode voltage for seven-phase inverters," in *Proceedings of the International Middle East Power Systems Conference, MEPCON*, Al Mansoura, Egypt, December 2015.

[19] S. Morimoto, Y. Takeda, K. Hatanaka et al., "Design and control system of inverter-driven permanent magnet synchronous motors for high torque operation," *IEEE Transactions on Industry Application*, vol. 29, no. 6, pp. 1150–1155, 2005.

[20] P. Zheng, P. Wang, Y. Sui, C. Tong, F. Wu, and T. Li, "Near-Five-vector SVPWM algorithm for five-phase six-leg inverters under unbalanced load conditions," *Journal of Power Electronics*, vol. 14, no. 1, pp. 61–73, 2014.

Hindawi
Complexity
Volume 2020, Article ID 6747985, 14 pages
https://doi.org/10.1155/2020/6747985

WILEY | Hindawi

Research Article

NECTAR-An Agent-Based Dynamic Task Allocation Algorithm in the UAV Swarm

Chao Chen, Weidong Bao, Tong Men, Xiaomin Zhu, Ji Wang, and Rui Wang

College of Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China

Correspondence should be addressed to Weidong Bao; wdbao@nudt.edu.cn

Received 2 August 2020; Accepted 29 August 2020; Published 16 September 2020

Academic Editor: Shi Cheng

The advancement of UAV technology makes the use of UAVs more and more widespread, and the swarm is the main mode of UAV applications owing to its robustness and adaptability. Meanwhile, task allocation plays an essential role in a swarm to obtain overall high performance and unleash the potential of each UAVs owing to the complexity of the large-scale swarm. In this paper, we pay attention to the real-time allocation problem of dynamic tasks. We design models for the task assigning problem to construct the constraints model and assigning objectives. In addition, we introduce a novel agent-based allocating mechanism based on the auction process, including the design for three kinds of agents and the cooperation mechanism among different agents. Moreover, we proposed a new algorithm to calculate the bidding values of UAVs, by which the messages passed between UAVs can be reduced. On the basis of the assigning mechanism, we put up with a novel agent-based real-time task allocation algorithm named NECTAR for dynamic tasks in the UAV swarm. Furthermore, we conduct extensive experiments to evaluate the performance of our NECTAR, and the results indicate that NECTAR is able to solve the real-time task allocation for dynamic tasks and achieve high performance of the UAV swarm.

1. Introduction

With the rapid development of artificial intelligence, high-end manufacturing, and robotics, the performance of Unmanned Aerial Vehicles (UAVs) is getting better and better. At present, UAVs have replaced humans in various complex situations, e.g., public security investigation, environmental protection detection, terrain exploration, and transmission line maintenance [1–3]. In addition, owing to UAVs' good stealth performance, strong autonomy, and reusability, UAVs are also widely used in battlefield environments by various countries. The main application mode of UAVs is the swarm, in which the UAVs would collaborate with each other to achieve operational performance.

It should be noted that the tasks undertaken by UAVs usually have strict time constraints in which the tasks' success depends not only on the execution but also on the time instants at which the execution finished. Therefore, the tasks should be assigned to suitable UAVs of the swarm as soon as possible to enhance their success probability. Consequently, there come two critical issues, one is how to find a suitable UAV in the swarm to avoid resource wasting, and the other is how to shorten the response time to promote the efficiency of the swarm.

Due to the characteristics of tasks, the allocation method plays an essential role in obtaining high performance for the swarm. Only by realizing reasonable task allocation can the cost of completing tasks be as small as possible, and coordination between different UAVs can be formed to achieve higher performance of the swarm. Besides, task allocation must try to save communication bandwidth as the communication capabilities of UAVs are usually limited [4]. To date, lots of allocation algorithms for UAVs have been studied. However, these algorithms do not sufficiently consider the characteristics of the UAV swarm, such as requirements of real-time response for dynamic tasks, low-bandwidth communication environment, and continuous task response.

In order to select proper UAVs from the UAV swarm to undertake tasks and make the response time short, we

comprehensively consider the characteristics of the UAV swarm, designing a novel agent-based assigning mechanism and, then, developing the auction-based task allocation algorithm for dynamic tasks.

The main contributions of our work can be summarized as follows:

(i) We designed models for the task-assigning problem in the UAV swarm and constructed the assigning constraints and objectives.

(ii) We propose an agent-based allocation mechanism based on the auction process. In addition, a novel calculation method of the bidding value was developed to reduce communication.

(iii) We investigate three selection strategies, the M strategy, P strategy, and D strategy, to determine the winner bidder.

(iv) We put up with a novel intelligent task allocation method, named NECTAR, for dynamic tasks in the UAV swarm.

The rest of the paper is organized as follows. We summarized the related works in Section 2. The problem description and model are given in Section 3. Section 4 is the designing of the agent-based mechanism. Also, we would give a detailed illustration of the NECTAR algorithm in Section 5. The experimental evaluation and analysis would be presented in Section 6. In Section 7, we concluded the paper with a summary and future work.

2. Related Work

Generally speaking, the models for task allocation problem for UAVs are mainly modeled into the following three categories: the mixed-integer linear programming model [5], dynamic network flow optimization model [6], and multiagent system [7]. These models have their own advantages and adaptation scenarios, but in general, the agent-based model has wider adaptability and better performance. Consequently, we choose the multiagent system to be the basic model. As for the solutions to the problem, many scholars have made attempts. We conclude them into two categories: the centralized task allocation method and distributed task allocation method.

Centralized solution schemes mainly include heuristic methods, such as genetic algorithm, particle swarm algorithm, simulated annealing algorithm, ant colony algorithm, and so on. Zhu et al. [8] focus on the problem of heterogeneous target assigning and UAVs' task sequence optimization, and they use a genetic algorithm to minimize the task execution time and UAVs' total consumption. As discussed in [9], the authors designed an improved particle swarm optimization method based on standard particle swarm optimization and evolutionary game theory, and they adopted a novel self-adaptive strategy in the task allocation process; the results show their advantages over other peers. Another heuristic method to solve the task allocation problem is the ant colony algorithm [10]; the authors considered the features such as being highly nonlinear, dynamic, highly adversarial, and multimodal, and they proposed a dynamic ant colony's labor division model based on the classic fixed response threshold model to finish task assignment. The simulation results identified its effectiveness in self-organization, flexibility, and real-time response to dynamic environments. Chun et al. [11] focused on a scenario where multiple air vehicles are required to prosecute geographically dispersed targets, aiming at solving the multirobot task allocation problems, and they developed an optimal task assignment algorithm based on a mixed-integer linear programming model. Also, the existence of the solution can be guaranteed. Other typical centralized algorithms such as wolf pack algorithm [12] and search algorithm [13] also show effectiveness in assigning tasks for multirobot systems. Owing to the fact that centralized methods have global information, they can obtain an ideal solution when dealing with task allocation problems in small-scale swarms. However, when the scale is large enough, it is also easy to be inefficient and difficult to obtain solution results quickly, and the reason is that the task assignment problem is an NP-hard problem. At the same time, since the centralized method is based on the status information of UAVs, communication bandwidth is required to meet the requirements of obtaining the real-time status of the UAV. However, the bandwidth of UAV swarms is one of the biggest limitations in current swarm applications. In addition, the centralized method also relies heavily on the central node, which makes this method not robust and adaptable.

When the scale of swarms becomes larger and the requirements for flexibility become higher and higher, the advantages of distributed methods are becoming more prominent. Thus, decentralized algorithms are receiving increasing attention. For example, Mehdi and Jonathan [14] improved the basic implicit coordination approach to finish the task allocation process, but the method is based on data synchronization among UAVs. However, synchronization is a difficult problem for swarms with limited communication bandwidth. Later, this work was extended by Whitten et al., and they considered coupled constraints to address complex mission characteristics in the allocation process. Besides, they designed the notion of pessimistic or optimistic bidding strategies and the relative timing constraints between tasks, and the results showed that it provided measurable performance improvement to the baseline CBBA [15]. As discussed in [16], the authors take limited bandwidth into consideration, they extended CBBA by duplicating cooperative tasks to modify the task list and added a judgment mechanism, and the simulation presented that their algorithm can achieve a better conflict-free allocation with fewer communications. In addition, Yang et al. [17] applied reinforcement learning in the task allocation for UAVs, they designed a networking scheme based on the expansion strategy, based on which the UAVs can improve the assigning through autonomous learning, and they conducted experiments on the UAV swarm to obtain optimized assigning in specific cases. Besides, many other algorithms such as sequential auctions [18], the human-agent collaboration method [19], and the bid-based grouping method

[20] are used. Though these distributed algorithms have solved the task allocation problem in UAVs, the communication limitation, real-time response, and continuous allocation are not sufficiently considered, making these methods not able to adapt to the actual environment.

In this research, we put up with a novel agent-based task allocation algorithm based on an improved auction mechanism to achieve real-time task allocation for dynamic tasks in the UAV swarm. The proposed agent-based assigning mechanism can quickly respond to the dynamic tasks with little communication, and it is able to continuously assign because the agents collaborate with each other based on their local status.

3. Problem Depictions

To study the dynamic task assignment problem, we mainly consider three kinds of UAV tasks, which are reconnaissance [21], striking [22], and damage assessment [23]. For dynamic tasks, they would arrive aperiodically, and the arrival time of them is unknown in advance. Besides, the tasks should be assigned as soon as possible owing to its deadline and the efficiency of the UAV swarm. Therefore, tasks should be assigned one by one rather than in a batch mode. Another fact that should not be ignored is that all UAVs would take off at the base station before its execution of tasks, and it should return to the base station before reaching its endurance time.

3.1. Models and Notations. In this paper, we mainly focus on the distributed task allocation method for the UAV swarm. Also, the tasks considered are uncoupled, and the reason is that coupled tasks can be decomposed by commanders into uncoupled tasks with other constraints such as time sequence and interaction order. Other features the tasks have include being nonpreemptive, deadline-sensitive, and aperiodic. We use a set $T = \{t_1, t_2, \ldots\}$ to represent the dynamic tasks. It should be noted that the count of tasks is not known because the tasks are appearing continuously as the confrontation progresses. For each task $t_i \in T$, it can be modeled as $t_i = \{ty_i, ta_i, td_i, tp_i, ts_i\}$, of which ty_i, ta_i, td_i, tp_i, ts_i denote t_i' type, arrival time, deadline time, position, and task supplementary information, respectively. Note that task type ty_i can be reconnaissance task, striking task, or damage assessment task.

We consider a set $U = \{u_1, u_2, \ldots, u_m\}$ of UAVs that have different capabilities. A UAV u_j of the swarm can be modeled by a collection of parameters, i.e., $u_j = \{uv_j, ut_j, ud_j, ltf_j, ltp_j, rsr_j, tft_j\}$, where uv_j, ut_j, ud_j, ltf_j, ltp_j, rsr_j, and tft_j are the u_j' velocity, ability type, duration time, finish time of the last execution task, position of the last execution task, remained striking resource, and total flying time, respectively. It is worth noting that each UAV has one or two kinds of ability among ability collection {*reconnaissance, striking, damage assessment*}. Besides, tft_j is the time from its take-off to its landing at the base station.

In Table 1, we summarize the main notations of this study for future reference. It should be noted that ltf_j and ltp_j represent the finish time and position of the last task in its task list. If a new task t_n has been assigned to u_j before the existing last task's completion, then the value of ltp_j would change to tp_n, and the value of ltf_j changes to f_{nj}. However, on the condition that no tasks have been allocated to u_j before the existing last task's completion, the position of u_j would be kept constant, which is because the UAV would keep hovering to save energy. Besides, we use rsr_j to denote the remained striking resource, which is used for task assignment. When the value of rsr_j equals to 0, it means UAV u_j can no longer undertake any striking tasks any more.

Let pt_{ij} be the preparation time of UAV u_j to execute task t_i, which refers to the flying time from u_j's last execution position to t_i' position, and the last execution position of u_j would be changed to t_i' position if t_i has been assigned to u_j. In addition, sp_{ij} and se_{ij} are used to denote the preparation start time and execution start time. Also, $U(t_i)$ refers to the UAV which undertakes task t_i. The allocation matrix $X = \{x_{ij}\}$ denotes the assigning relations between tasks and UAVs. In addiyion, the value of element x_{ij} should be "1" or "0", which means task t_i is allocated to UAV u_j or not.

3.2. Constraints. As is assumed before, the tasks discussed in this paper are not splittable, and a task can only be assigned to one UAV. On the other hand, different tasks usually have different position requirements; thus, each UAV can only undertake one task at any time instant. Therefore, we can conclude the constraint C_1 as follows:

$$C_1: \begin{cases} \sum_{j=1}^{m} x_{ij} = 0 \text{ or } 1, \quad i \in [1, 2, \cdots], \\ x_{ij} \cdot x_{kj} = 0, \quad \text{if } [sp_{ij}, f_{ij}] \cap [sp_{kj}, f_{kj}] \neq \varnothing. \end{cases} \tag{1}$$

For the preparation start time sp_{ij}, execution start time se_{ij}, and finish time f_{ij}, they are determined by the t_i's arrival time and finish time of u_j's last task. As is depicted in Figure 1, the preparation start time sp_{ij} can be calculated as follows:

$$sp_{ij} = \max\{ta_i + tb_i,\ f_{kj}\}, \tag{2}$$

where tb_i represents the time spent for the auction process. Furthermore, the execution start time se_{ij} can be determined by formula (3), and the finish time f_{ij} can be calculated by (4):

$$se_{ij} = sp_{ij} + pt_{ij}, \tag{3}$$

$$f_{ij} = se_{ij} + e_{ij}. \tag{4}$$

Whether a task t_i can be assigned to a certain UAV u_j or not is mainly determined by time constraint and ability constraint, which can be concluded as C_2:

$$C_2: \begin{cases} f_{ij} \leq td_i, & \text{if } x_{ij} = 1, \\ \Phi(t_i) \in ut_j, & \text{if } x_{ij} = 1, \end{cases} \tag{5}$$

where $\Phi(t_i)$ means the required ability of task t_i. Time constraint mainly refers to the fact that the finish time f_{ij} must satisfy t_i' deadline. Ability constraint mainly indicates that the UAV must have the ability to process task t_i.

Table 1: Definition of main notations.

Notation	Definition
t_i	The i^{th} task in task set T
ty_i, ta_i, td_i, tp_i, and ts_i	Task t_i' type, arrival time, deadline time, position, and task supplementary information
u_j	The j^{th} UAV in UAV set U
uv_j, ut_j, ud_j, ltf_j, ltp_j, rsr_j, and tft_j	UAV u_j' velocity, ability type, duration time, finish time of the last execution task, position of the last execution task, remained striking resource, and total flying time
pt_{ij}	The preparation time of UAV u_j to execute task t_i
sp_{ij} and se_{ij}	The preparation start time and execution start time of task t_i on UAV u_j
e_{ij} and f_{ij}	The execution time and finish time of t_i on u_j
$U(t_i)$	The UAV to which t_i is assigned
x_{ij}	x_{ij} is "1" if t_i is assigned to u_j; otherwise, x_{ij} is "0"

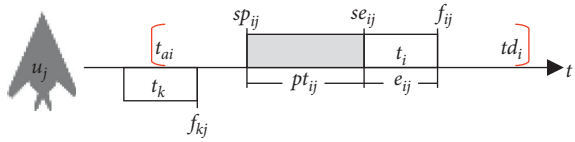


Figure 1: The task assigning parameters.

Another constraint comes to the striking tasks, and they need the remained striking resource to be not null at the time of task assignment. Also, the constraint is modeled as

$$C_3: \begin{cases} x_{ij} = 0, & \text{if } t_i = \text{striking}, rsr_j = 0, \\ x_{ij} = 1 \text{ or } 0, & if\ t_i = \text{striking}, rsr_j \neq 0, \end{cases} \tag{6}$$

and note that such constraint is only considered in t_i's assigning process because rsr_j may be equal to 0 after the task's execution.

3.3. Assigning Objectives. For the UAV swarm, it should complete as more dynamic tasks as possible to realize swarm efficiency. Besides, each UAV should try to increase the proportion of execution time in the total flying time as much as possible to realize a single UAV's efficiency. Consequently, we take the task success ratio and execution time ratio as two main assigning objectives.

Task success ratio:

$$\max\left\{\frac{\sum_{i=1}^{|T|}\sum_{j=1}^{m} x_{ij}}{|T|}\right\}. \tag{7}$$

Execution time ratio:

$$\max\left\{\frac{\sum_{i=1}^{|T|}\sum_{j=1}^{m} x_{ij}\cdot e_{ij}}{\sum_{j=1}^{m} tft_j}\right\}. \tag{8}$$

The abovementioned equations indicate that our agent-based assignment method aims at allocating more tasks to UAVs for successful execution within complex constraints and realizes higher UAV utilization.

4. Agent-Based Assigning Mechanism Design

On the designing of the agent-based assigning method for the UAV swarm, we mainly try to employ an auction mechanism [24] to complete task allocation. The auction mechanism allows agents to cooperate to finish the task assignment process in a distributed way, which makes it ideal for the problem of UAV task allocation. Note that, although the bidding process is managed by the central management agent, there is no central control [25]. More importantly, we devise the calculation of bidding values to local agents obeying global uniform rules, by which reducing the communication load among agents.

4.1. Agent Design. To realize agent-based task assigning, three kinds of agents are designed, i.e., the manager agent, task agent, and UAV agent. Each of them works based on its own rules, and they would cooperate with each other to finish the auction process. We use a triple tuple $<M^A, T^A, U^A>$ to represent the agents used in this paper.

(i) M^A represents the manager agent. The manager agent is yield with the construction of the swarm and would exist all the time. In addition, the manager agent is responsible for the management of all task agents and UAV agents.

(ii) $T^A = \{t_1^A, t_2^A, \ldots\}$ is a task agent set, and t_1^A denotes the i^{th} task agent in T^A. A task agent is mainly responsible for the auction and execution of the task. For each task agent, it would be constructed with the arrival of its corresponding task and die out with the finish. Note that t_1^A locates on a UAV which is responsible for the task t_i, and the system resources occupied by t_1^A would be reclaimed by the UAV after the finish of t_i. As for which UAV t_1^A is located, it depends on how the mission appears. If the task is discovered by a UAV u_j and approved by the commander, t_1^A would locate on u_j; if the task is discovered by the commander and assigned to the UAV u_k, t_1^A would locate on u_k.

(iii) $U^A = \{u_1^A, u_2^A, \ldots, u_m^A\}$ is a UAV agent set, and u_j^A represents the j^{th} UAV agent in U^A. A UAV agent is responsible for the management of UAV resources and bidding for tasks. For each UAV agent, it would exist all the time with the related UAV. Besides, each

UAV agent would update its status information and report to the manager agent in the following conditions: (1) after the construction of the UAV agent; (2) after a striking task has been assigned to it; and (3) encountering an abnormal situation.

4.2. Design of the Auction Mechanism. In the auction process, there are mainly two kinds of roles, i.e., announcers and bidders. The work of announcers would be mainly undertaken by task agents, and that of bidders would be mainly finished by UAV agents. Consequently, we would focus on the interactions among different agents to illustrate the auction mechanism.

Considering the problem of task assigning in the UAV swarm, the key point is to allocate each arriving task to a proper UAV. In this work, we use an auction mechanism to determine which UAV would be the proper one for a certain task. As is the case in auction activities, the first step is to analyze task requirements and select candidate UAVs, and we conclude these works into the preparation stage. The following stages are the announcing, bidding, awarding, and execution stage. Figure 2 depicts the basic activities and interactions of the agents, and we would give a more detailed illustration for the auction mechanism as follows.

4.2.1. Preparation Stage. In this stage, there are mainly three types of work. The first is that UAV agents report basic capacity information to the manager agent when coming to the conditions mentioned in the Agent Design part. The second is that the task agent sends basic information to the manager agent when the task agent is generated. The third is that the manager agent finishes the matching process and sends information of candidate UAVs to the task agents.

4.2.2. Announcing Stage. The main work of this stage is announcing to candidate UAVs the task agent would receive the information of candidate UAVs and send basic task information to these matched UAVs for the invitation. As for UAV agents, they would receive the bidding invitation and task information.

4.2.3. Bidding Stage. In this stage, each UAV agent would calculate the bidding value based on its status parameters (including position, earliest finish time, and duration time) and task information. Also, the calculation should obey the predefined common rules. After getting the bidding value, UAV agents would send the values to the task agent for bidding. It should be noted that the bidding process does not conflict with the task execution, which means that a UAV can bid to a new task when executing an assigned task. The reason is the announcing task is to be executed in the future time and has no influence on the current execution process.

4.2.4. Awarding Stage. After the task agent has received bidding values from bidders, it would select a proper UAV to award the contract according to predefined selection strategies. Also, the complete task information would be sent by the task agent to the winner UAV agent.

4.2.5. Execution Stage. In this stage, the winner UAV would execute the task and report the result to the task agent after the execution.

In Figure 3, we give an example of assigning a task to a proper UAV based on the auction mechanism. In this example, task t_i is to be assigned, and we assume that there are three UAV agents: u_1^A, u_2^A, and u_3^A. Task agent t_i^A firstly sends its basic task information to the manager agent M^A. Then, M^A would select UAVs that satisfy t_i's requirements to be the matched UAVs, and u_1, u_2, u_3 are the selected ones in this example. After the matching process, M^A sends the UAVs' information to t_i^A. Thirdly, t_i^A sends bidding invitation and necessary task information to u_1^A, u_2^A, and u_3^A, and the UAVs would calculate the bidding values about t_i on their local devices. Next, the three UAVs bid to t_i by sending their own bidding values to t_i^A. After receiving the bidding values, t_i^A selects u_2^A as the winner UAV based on bidding values and selection strategy. Finally, t_i^A sends the complete task information to u_2^A and awarding to it. After the task assigning, task execution and reporting would be performed by u_2^A.

4.3. Bidding Value. In the auction mechanism mentioned before, the task agent selects a proper UAV based on the bidding values. Consequently, the calculation of bidding values is significant to the efficiency. In the bidding process, whether a candidate UAV can satisfy the deadline is the most important factor, and we describe this effect by a step function.

$$\varphi\left(td_i - f_{ij}\right) = \begin{cases} 1, & \text{if } td_i - f_{ij} > 0, \\ 0, & \text{if } td_i - f_{ij} \le 0. \end{cases} \tag{9}$$

Another factor that should be taken into consideration is the execution start time, and it should be as early as possible. Besides, the preparation time would be better to be short. In addition, it would be better to have a short execution time. Therefore, we calculate the bidding values of UAV u_j to task t_i by the following equation:

$$b_{ij} = \frac{\varphi\left(td_i - f_{ij}\right) \cdot e_{ij}^{-1}}{pt_{ij} \cdot \left(se_{ij} - ta_i\right)}. \tag{10}$$

4.4. Selection Strategies. In the auction process, the task agent would select a bidding UAV agent to award a contract if there is, at least, one bidding value which is not 0. Note that different selection strategies may have distinctive performances. In this work, we present three selection strategies, which are the M strategy, D strategy, and P strategy. Also, we would give detailed illustrations later.

When task t_i ($i = 1, 2, 3, \ldots$) arrives, there would be a UAV responsible for its auction work, and we use *AN* (short for announcer) to denote the UAV. As for the UAVs bidding to t_i, we use $BID = \{bid_j, j = 1, 2, \ldots\}$ to represent. What we need to pay attention to is that when the bidding value is 0,

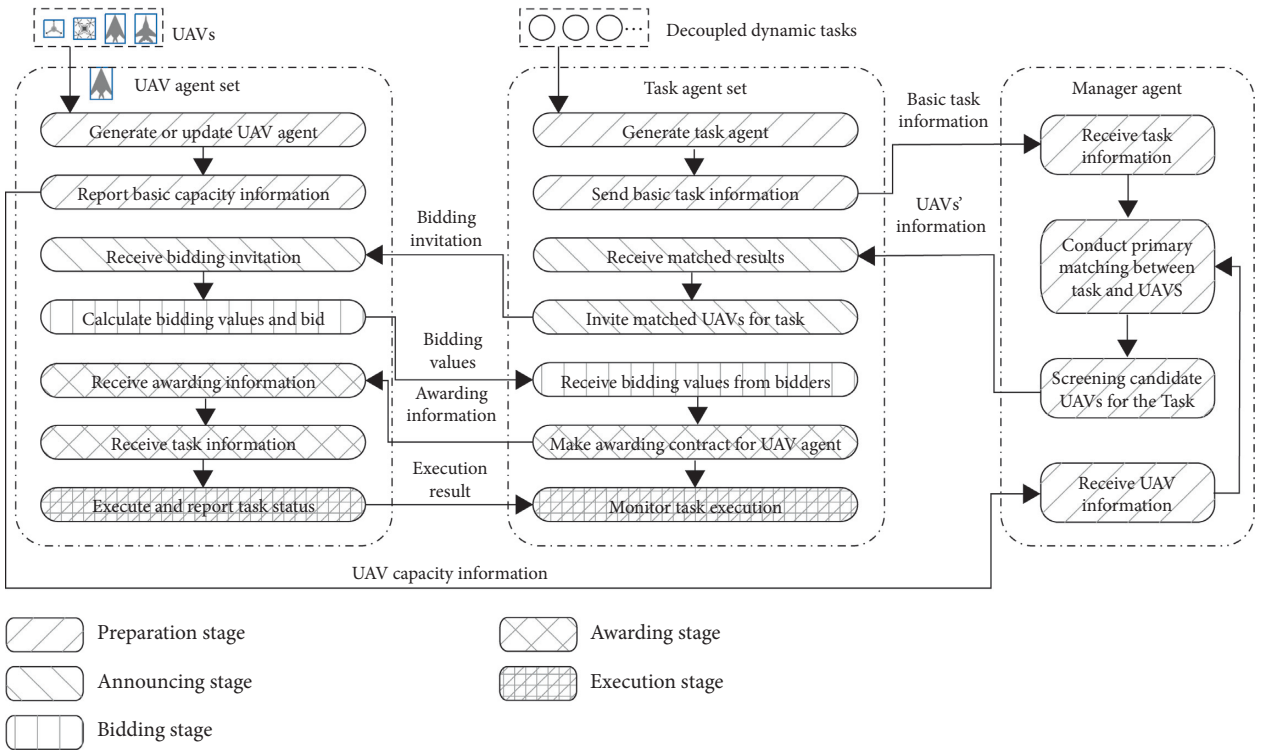


Figure 2: The basic activities and interactions of agents.

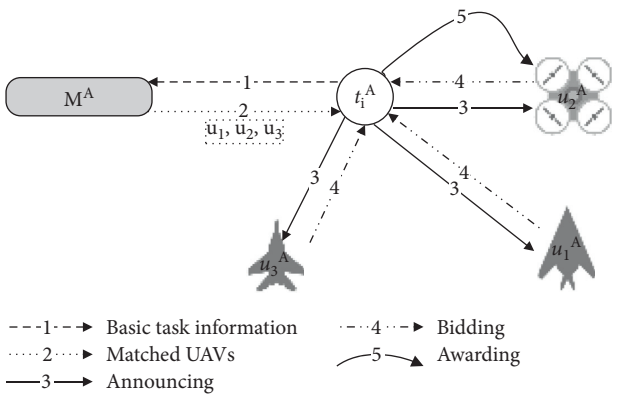


Figure 3: An example of task assigning.

the corresponding UAV will not be treated as a valid bidder, and it will not be added into the set *BID*. Besides, BV = $\{bv_j, j = 1, 2, \ldots\}$ is the bidding value set, in which bv_j represents the bidding value of bidder bid_j for *AN*. It should be noted that when there is only one bidder in *BID*, it would be awarded directly, so the selection strategy is only applicable in the condition of more than one bidder.

4.4.1. M Strategy. Under this strategy, the *AN* selects the bidder with maximal bidding value. If bid_k is selected, it must meet the following constraint:

$$\forall bid_j \in BID, \quad bv_k \geq bv_j. \tag{11}$$

4.4.2. P Strategy. Under this strategy, the *AN* chooses a bidder according to probability policy. The winning probability wp_j of bidder bid_j would be determined by

$$wp_j = \frac{bv_j}{\sum_{i=1}^{|BID|} bv_i}. \tag{12}$$

To select the bidders based on probability, we let *wp* be a random number, and wp ∈ (0, 1). Without loss of generality, we let $wp_0 = 0$. When the value of *wp* satisfies equation (13), bidder bid_k is chosen as a winner.

$$\sum_{m=0}^{k-1} wp_m < wp \leq \sum_{n=0}^{k} wp_n. \tag{13}$$

4.4.3. D Strategy. Considering that the distance between the task and the UAV is a key factor that affects the execution efficiency, the D strategy tries to select a bidder with the condition that the bidding value is larger than 0 and has the shortest distance among all bidders. It is worth noting that the bidders should send the value of distance with the bidding value to the announcer. We let d_j be the distance between the *AN* and bidder bid_j. If bid_k is selected, then the following constraint must be satisfied:

$$\forall bid_j \in BID, \quad d_k \leq d_j. \tag{14}$$

5. NECTAR Method

Based on the mechanism mentioned above, we propose a novel agent-based dynamic task allocation algorithm, NECTAR, for the UAV swarm. In this section, we would give detailed illustrations for NECTAR by presenting the algorithms of the manager agent, task agents, and UAV agents.

In Algorithm 1, we present the pseudocode of the algorithm for the manager agent. The manager agent initializes its operation in line 1, after which the existing UAV agents would be added into the agent set U^A. In line 3, the function *Waiting( )* is running in an independent thread, and the manager agent would jump out of the waiting state only when there are requests from task agents or reports from UAV agents. Line 4 means that there is a new request from task agent t_i^A. Also, the manager agent would let the candidate UAV agent set to be null in line 5. Then, the manager agent polls all the agents in U^A (line 6). If the task is not a striking task, all the UAV agents satisfying task t_i's ability requirement would be added into set $U^A_{candidate}$ (line 7). However, if the task is a striking task, the selected agents must satisfy additional resource constraints, i.e., the rsr_j should not be 0 (lines 9–10). Line 12 means that there is a new report from UAV agent u_m^A. If the agent is newly constructed and not in U^A, the manager agent would add it to U^A and update the related state. Otherwise, the status information of u_m^A would be updated directly (lines 12–15). After the response for a request or a report, the manager agent would return to function *Waiting( )* (line 16). Also, the manager agent would jump out of the waiting state and start processing at line 4 when the next request or report is coming.

The pseudocode of the algorithm for the task agent is given in Algorithm 2. After the construction of the task agent, it would initialize and analyze the task information (line 1). Lines 2 and 3 show that the task agent would send the task requirements to the manager agent and receive the candidate UAV agents. It should be noted that the basic requirements are mainly the task type and its additional information, such as resolution and firepower type. The task agent would announce to the candidate UAV agents for the task (lines 4-5). In lines 6 and 7, the task agent would initial the valid bidding value set to be an empty set, after which the bidding value of each bidder would be received. If a UAV agent has the bidding value bigger than 0, the value would be added to the valid bidding value set *BidValue* (lines 9–10). In lines 11 and 12, the task agent would select a winner bidder based on selection strategies. However, if the set *BidValue* is null, that means there is no UAV agent that can satisfy task demands, and the task should be rejected (lines 13-14).

Algorithm 3 depicts the pseudocode of the algorithm for UAV agents. Line 1 means that the UAV agent is constructed, and it would initialize. Then, the UAV agent would acquire status parameters and report to the manager agent for registration, and the reporting information mainly includes abilities and remaining resources. In line 3, the function *UagentWaiting()* would run in an independent thread, and the UAV agent would jump out of the waiting state in the following three conditions: bidding invitation from task agents, awarding message from task agents, and the UAV encounters an abnormal situation. Note that abnormal situations mainly refer to device errors which lead to the loss of abilities. Lines 5-7 describe the first situation that causes the UAV agent to jump out of the waiting state. The UAV agent would analyze the task information and calculate the bidding value, after which the bidding value is to be sent to the corresponding task agent. Besides, the awarding message would also lead the UAV agent to jump out of the waiting state (lines 8–12). The UAV agent would add the task to the UAV task list and update status parameters. If the task is a striking task, the UAV agent should lock the striking resource and report to the manager agent. Another situation that would lead to jumping out of the waiting state is an abnormal situation. If this situation occurs, the UAV agent would acquire the device states and report to the manager agent M^A (lines 13–15).

As illustrated above, NECTAR would allocate the task at its arrival rather than allocate the tasks in batch mode, so the NECTAR achieves real-time allocation for dynamic tasks. In addition, the calculation of bidding values would locate on the UAV agent rather than the manager agent, by which the status messages passed between each other will be greatly reduced, so it is able to be used in a limited bandwidth environment.

6. Experimental Evaluation

As illustrated in Section 4, there are four kinds of selection strategies. Thus, four kinds of task assigning algorithms can be generated by adopting different bidder selection strategies, i.e., NECTAR-M, NECTAR-P, and NECTAR-D. It should be noted that NECTAR-M employs the M strategy when selecting the winner bidder, and NECTAR-P and NECTAR-D adopt the P and D strategy, respectively. In addition, we would compare them with a baseline algorithm- RANDOM. For the sake of avoiding central control to reduce communications, RANDOM uses an agent-based mechanism (similar to NECTARs) to get candidate UAVs and randomly chooses the winner UAV to allocate the task.

To verify the effectiveness of our proposed algorithms and ensure the repeatability of the experiments, we conduct simulation experiments to evaluate the performance of NECTAR. Also, the performance metrics by which we evaluate algorithms' efficiency mainly include the following.

6.1. Task Success Ratio (TSR). It can be defined as *TSR* = Total count of successful tasks/Total count of arrived tasks, and the mathematical expression can be found in (7).

6.2. Execution Time Ratio (ETR). The definition of the *ETR* can be described as *ETR* = Total execution time of all tasks/Total flying time of all UAVs, and the mathematical expression can be found in (8).

(1) Initialize and let UAV agent set $U^A \longleftarrow NULL$;
(2) $U^A \longleftarrow$ existing UAV agents;
(3) *Waiting( )*;
(4) **if** *There is a new request from* t_i^A **then**
(5) Let $U^A_{candidate} \longleftarrow NULL$;
(6) **foreach** u_j^A *in* U^A **do**
(7) **if** $\Phi(t_i)\, u_j^t$ *and* $t_y^j \neq striking$ **then**
(8) Add u_j^A into $U^A_{candidate}$;
(9) **else if** $\Phi(t_i)\, u_j^t$ *and* $rsr_j > 0$ **then**
(10) Add u_j^A into $U^A_{candidate}$;
(11) Send $U^A_{candidate}$ to t_i^A
(12) **else if** *There is a new report from* u_m^A **then**
(13) **if** u_m^A is not in U^A **then**
(14) Add u_m^A into U^A;
(15) Update related status of u_m^A in U^A;
(16) Return to *Wainting( )*;

Algorithm 1: The algorithm for the manager agent.

(1) Initialize and analyze task information;
(2) Send basic task requirements to M^A;
(3) Receive candidate UAV agent set $U^A_{candidate}$;
(4) **foreach** u_j^A *in* $U^A_{candidate}$ **do**
(5) Send announcement information to;
(6) Let valid bidding value set *BidValue* $\longleftarrow$ *NULL;*
(7) Receive bidding value from UAV agents;
(8) **foreach** u_j^A *in* $U^A_{candidate}$ **do**
(9) **if** $bid_j > 0$ **then**
(10) Add bid_j into set *BidValue*;
(11) **if** *BidValue is not null* **then**
(12) Select a winner bidder Bid_{win} based on the selection strategies;
(13) **else**
(14) Reject task;

Algorithm 2: The algorithm for the task agent.

(1) Initialize and analyze UAV status;
(2) Acquire parameters of status;
(3) Report status parameters to the manager agent M^A;
(4) *UagentWaiting()*;
(5) **if** *There is a bidding invitation from* t_i^A **then**
(6) Analyze the task information;
(7) Calculate bidding value and send value to t_i^A;
(8) **if** *There is an awarding message* **then**
(9) Add task to UAV task list;
(10) Update status parameters;
(11) **if** *Awarded task is a striking task* **then**
(12) Report remained striking resource to M^A;
(13) **if** *The UAV encounters an abnormal situation* **then**
(14) Acquire device ability parameters;
(15) Report status parameters to M^A;
(16) Return to *UagentWaiting()*;

Algorithm 3: The algorithm for the UAV agent.

6.3. Simulation Parameters. For the parameters mentioned above, we give a detailed setting as follows:

(i) The tasks arrive dynamically, following the Poisson distribution with the average internal time $1/\lambda$.

(ii) Parameter *BaseDeadline* is used to set the deadline of task t_i, and the formula is as follows:

$$td_i = ta_i + \text{BaseDeadline}, \tag{15}$$

where parameter ta_i represents the arrival time of t_i and *BaseDeadline* follows a uniform distribution in the interval $[40, 40 + a \times 10]$.

(iii) The initial number of UAVs is set to be 150.

(iv) The tasks are located in a rectangle area, the length of which is 3000 m and the width is 2000 m. At the time instant of 0, the UAVs are randomly distributed in the mission area.

(v) The UAVs' maximum velocity is set to be *60* m/s according to current technologies [26].

(vi) The set of abilities possessed by each UAV is a true subset of set {*reconnaissance ability, striking ability, damage assessment ability*}. For the striking resources, the initial number is 10, which means that the maximum number of strike tasks the UAVs can undertake is 10. Also, the UAVs of the striking type would return to their base when the striking resources have been used up.

(vii) The execution time follows a uniform distribution pattern, and the value of execution time is determined by their types. For reconnaissance tasks, the value of e is between [15, 25], striking tasks e~[25, 35], and assessment tasks e~[35, 45].

We mainly carry out four groups of experiments to study the effectiveness of the proposed algorithms, NECTARs, and analyze the impact of time, task arrival rate, task deadline, and the number of UAVs on the performance. The intervals of these parameters are listed in Table 2.

Table 2: Parameters for experiment studies.

Parameter	Value (fixed) – (varied)
Experiment Time	(500) – (200 900)
λ	(3) – (1,2,3,4,5,6,7)
a	(4) – (1,2,3,4,5,6,7)
UAV numbers	(150) – (60, 90, 120, 150, 180, 210, 240)

6.4. Performance with Time Variation. First, we study the performance of NECTAR algorithms when time varies. It is noted that time mainly affects the number of striking UAVs because strike resources are consumable, so the later the time, the fewer the striking UAVs available. In this group of experiments, λ equals 5, the value of a is 4, the number of UAVs is 150, and the statistical time varies from 200 to 900. The result can be seen in Figure 4.

From Figure 4(a), we can see that all algorithms have achieved a task success ratio of over 80% in the beginning, which can verify the effectiveness of the agent-based auction process. By comparing the *TSR* of the four algorithms, we can find that NECTAR-M has the highest success rate, and RANDOM achieves the lowest success rate. As for the reason that RANDOM has the lowest *TSR*, we attribute this to the random selection strategy. Due to the strategy, it will select a winner UAV randomly from candidate UAVs who meet the task's deadline. However, the selected UAV may need long transition time and wait a period of time before the start of the task's execution, which greatly affects the efficiency of the UAV, resulting in resource wasting, so the *TSR* is low. From the comparison of NECTAR-D and RANDOM, we would see that the performance gap between them is about 6%, so we can conclude that the distance is a main factor affecting the task assigning efficiency. Consequently, it would be easy to understand that NECTAR-D has slightly better performance than NECTAR-P, and the reason is that NECTAR-P only has a larger probability to select the UAVs with larger bidding value, but NECTAR-D would choose the nearest UAV to undertake the task, leading to the time spent for transition being the shortest. As for the reason that NECTAR-M has the highest *TSR*, it can be concluded to the employment of maximum bidding values selection strategy, which takes the distance, execution time, and UAV waiting time into consideration, comprehensively. Therefore, it would choose a UAV that is relatively suitable for the task and obtains higher effectiveness. From the comparison between NECTAR-M and NECTAR-D, we can find NECTAR-M achieves higher performance than NECTAR-D for about 4%, which indicates the waiting time and execution time would be another factor that should not be ignored in the assigning process.

Another phenomenon observed in Figure 4(a) is that the performance of all algorithms would decrease with time variations. The reason is that the striking resource is limited, and more striking tasks may be rejected due to the exhaustion of striking resources. Furthermore, the performance gap between NECTAR-M and NECTAR-D has become smaller and smaller over time. The reason can be attributed to the rejection of striking tasks, which makes the advantage of NECTAR-M can only be reflected in reconnaissance tasks and damage assessment tasks, leading to the gaps becoming smaller.

We can observe from Figure 4(b) that NECTAR-M achieves the highest *ETR* among the four algorithms, and the reason is that the M strategy would select the most suitable UAV to reduce the idle time and flying distance in a comprehensive way, and thus, the execution time ratio would be higher. The performance of other algorithms on the *ETR* is NECTAR-P, NECTAR-D, and RANDOM in order, and the explanation of this fact is similar to that on the *TSR*. An interesting observation is that the *ETR* of NECTAR-M, NECTAR-P, and NECTAR-D would slightly decrease with time variation, and the descending speed would increase after the time 500s. That is because after 500s, the resources of some striking UAVs are exhausted and the UAVs would return to the base, the available striking UAVs become less, so the UAVs undertaking striking tasks will

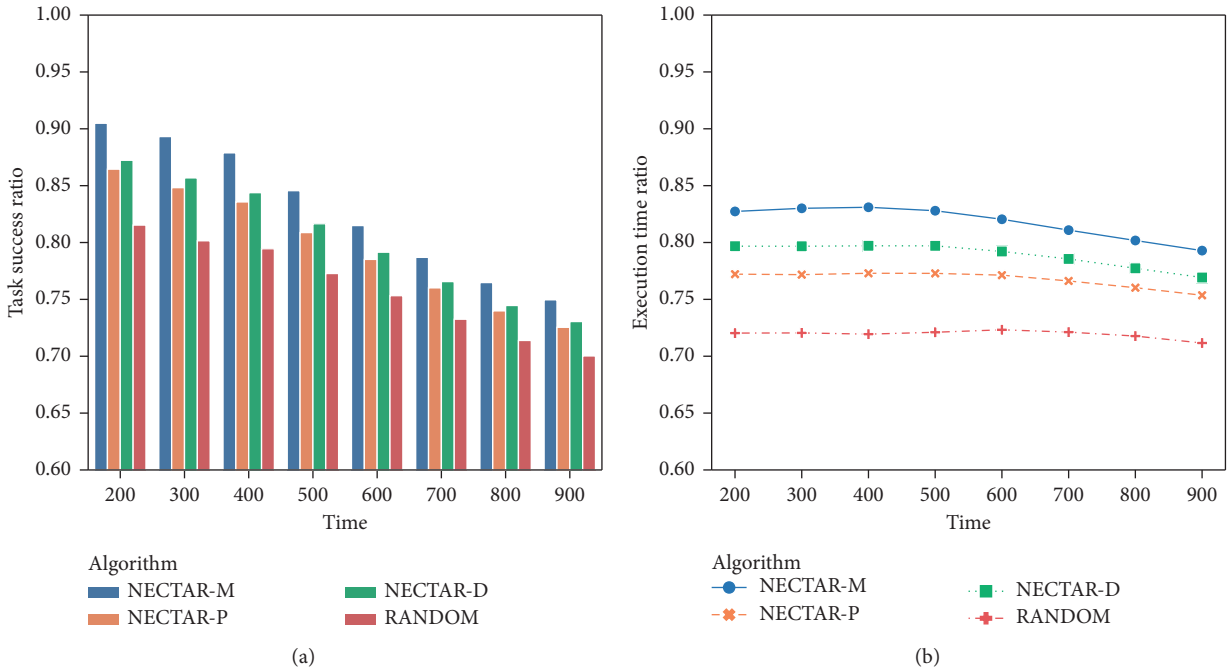


Figure 4: Performance with time variation.

have to fly longer distances, resulting in a lower *ETR*. However, the *ETR* of RANDOM is to first keep stable and, then, fall slightly. The reason for the smaller decline on the *ETR* is that RANDOM does not save the transition time, waiting time, and execution time in the beginning, so the reduction in the number of available UAVs has less impact on RANDOM.

We can conclude from Figure 4 that the proposed agent-based auction mechanism is effective in assigning tasks to the UAV swarm, and the main factors affecting the assigning efficiency are the transition distance, waiting time, and execution time. In addition, the performance verifies the effectiveness of the designed NECTARs.

6.5. Performance Impact of the Task Arrival Rate. In this group of experiments, we pay attention to the impact of the arrival rate on the performance of NECTARs and RANDOM. The parameter λ varies from 2 to 8 with an increment of 1. Note that the parameter λ mainly affects the number of arrival tasks in a certain unit of time. Also, the experiment results are presented in Figure 5.

It can be seen from Figure 5(a) that when the value of λ is 2, all algorithms have achieved a 100% *TSR*. The reason is that the number of arriving tasks is small and all tasks can be successfully executed as the UAV resources are sufficient. When the value of λ is larger than 3, the task success ratio of the four algorithms will decrease with the increment of λ. This can be explained by the fact that when the task arrival rate becomes large enough, all UAVs have reached a heavy burden, and they have no choice but to reject some tasks due to the limited resources, and thus, the *TSR* would decrease.

An interesting phenomenon found in Figure 5(a) is that, with the increment of λ, the performance gap between NECTARs and RANDOM would first become larger and, then, smaller. The reason is that as the number of tasks increases, NECTAR's advantages become more and more manifest, so the performance gap becomes larger. Also, when the task load reaches a certain level, the advantage of NECTARs over RANDOM would keep stable as the potential of UAVs has been realized, but the total tasks would keep increasing with λ; thus, the proportion of the advantage in total tasks would decrease, so the performance gap becomes smaller.

From Figure 5(b), we can observe that the *ETR* would increase with the increment of λ. At first, it would increase rapidly, which is because the UAVs are in light load, and more tasks would help UAVs improve the utilization. However, when the UAVs are in heavy load, the improvement of utilization would slow down or nearly to be 0, so the *ETR* would grow slowly or keep stable.

It can be concluded from Figure 5 that the parameter λ mainly affects the workload state for UAV swarms, and the proposed NECTARs can adapt to the environment of different workloads.

6.6. Performance Impact of Task Deadline. We study the impact of the task's deadline on the performance of the NECTARs in this group of experiments. The parameter a varies from 1 to 7 with an increment of 1. It should be noted that the parameter a mainly affects the deadline, and when the value of a becomes larger, the deadline of tasks would be looser. Also, the results are depicted in Figure 6.

We can observe from Figure 6(a) that the *TSR* of NECTARs and RANDOM would grow with the increasing of parameter a's value. This is because when the task deadline is looser, tasks can be more flexibly assigned to

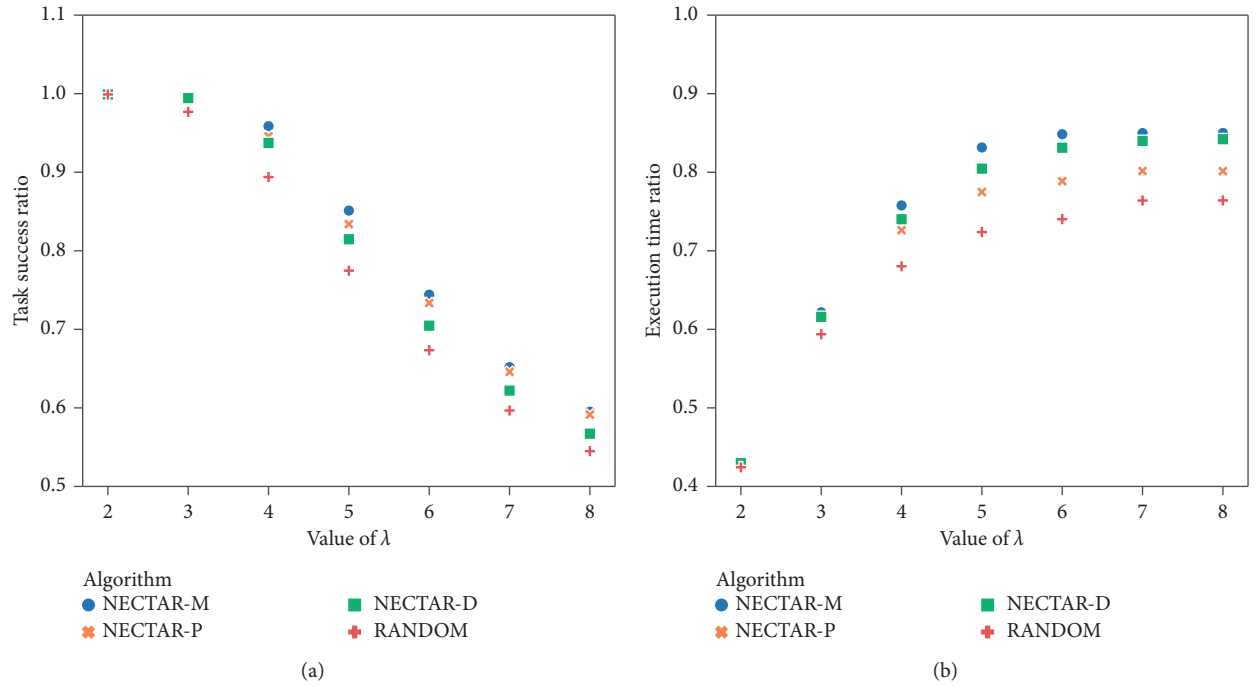


Figure 5: Performance impact of the task arrival rate.

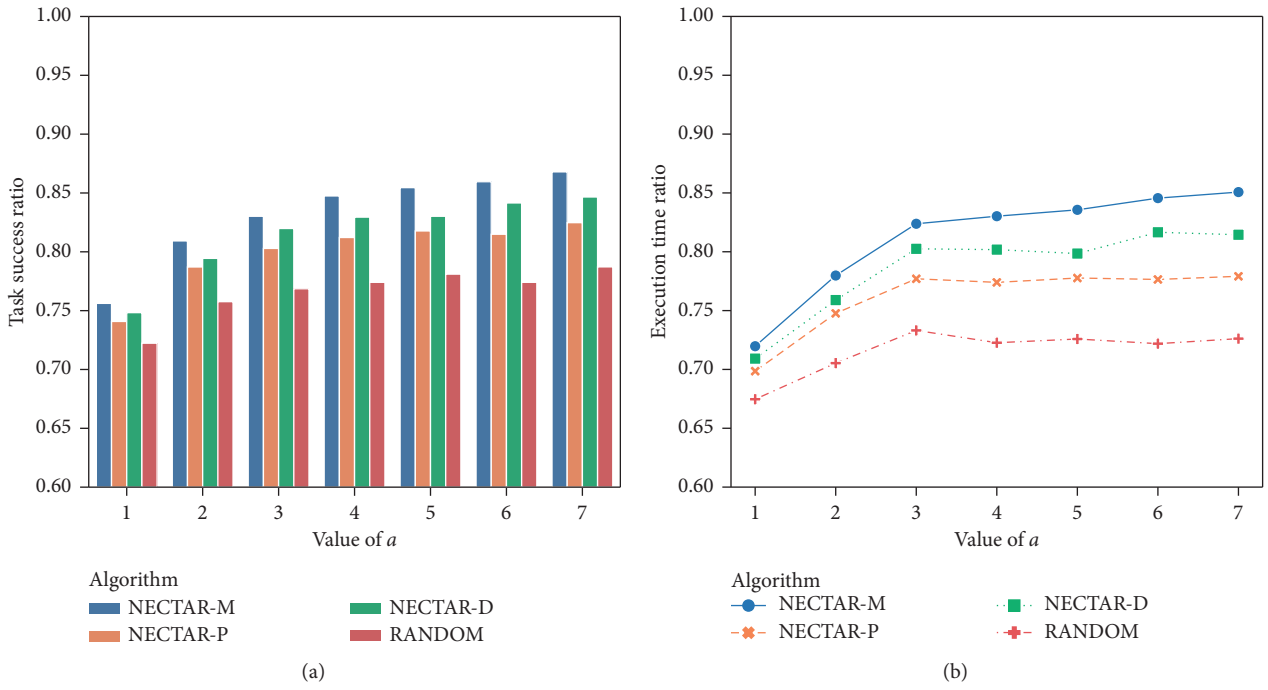


Figure 6: Performance impact of task deadline.

UAVs, which improves the utilization of UAVs. Another fact that can be concluded is that the growth of the *TSR* is fast at first and slows down later. The reason is that the success of tasks is mainly affected by the number of available UAVs, looser deadline would help to promote the utilization of a single UAV, but the promotion is limited. Consequently, the growth of the *TSR* would slow down and the *TSR* would keep stable at last. Besides, we can observe from Figure 6(a) that the growth of NECTAR-M is the largest, and this verifies the efficiency of the proposed mechanism. We conclude the reason to be that NECTAR-M has taken distance and time into consideration in a

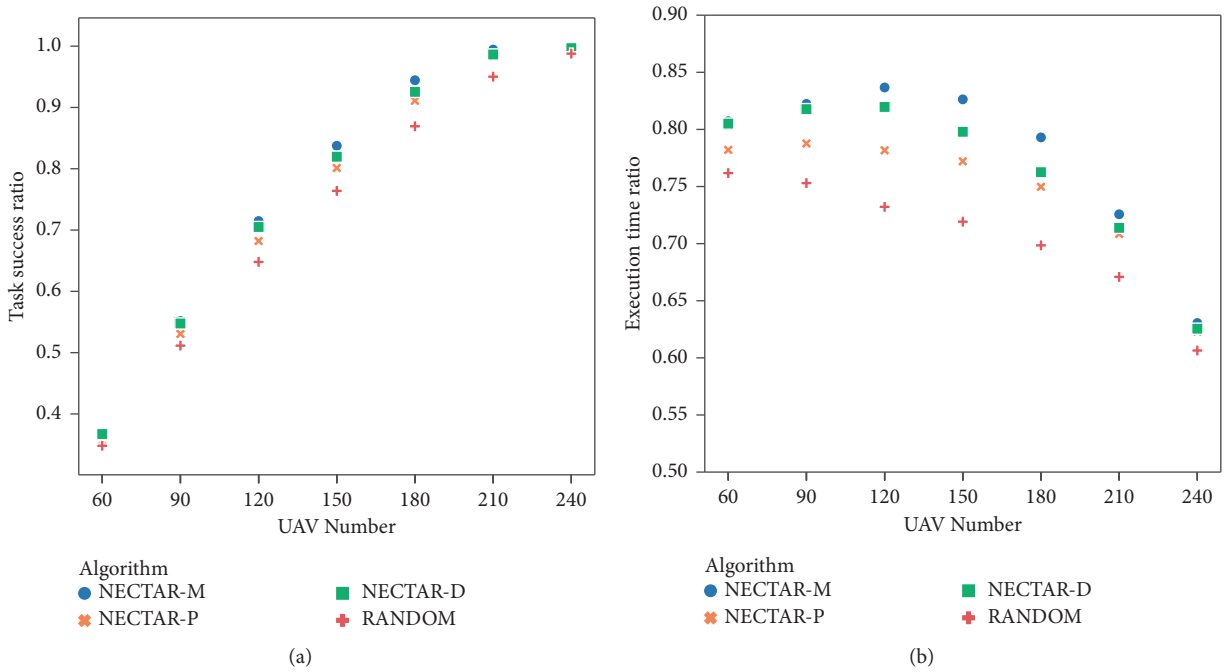


Figure 7: Performance impact of the UAV number.

comprehensive way, so the improvement of UAV's utilization would be more significant.

From Figure 6(b), we can find that the *ETR* of NECTARs and RANDOM would grow with an increment of *a*. Another fact is that the performance gap on the *ETR* between different algorithms would become larger with looser deadline. Also, the explanations would be similar to those of the *ETR*.

It can be summarized from Figure 6 that when tasks have looser deadline, the performance of assigning algorithms would get improved, but the improvement is limited. In addition, our proposed NECTARs achieve better improving on both the *TSR* and *ETR*.

6.7. Performance Impact of Task Deadline. In this group of experiments, the impact of the UAV number would be studied. Also, the UAV numbers mainly affect the available UAV resources in the swarm. The results are presented in Figure 7.

As depicted in Figure 7(a), the *TSR* of NECTARs will continue to grow with the increase of the UAV number until it reaches 100%. The reason is that there are more UAVs that can respond to tasks, so more tasks can be successfully executed. Besides, the performance gap on the *TSR* between NECTARs and RANDOM would first become larger and, then, smaller. The explanation is that the tasks are arriving dynamically and must be responded to at its arrival, when the number of UAVs is too small, the range of options is small, and the optimization effect is limited, so the advantages of NECTARs are not significant. While the number of UAVs increases, the NECTARs would achieve obvious performance advantages over RANDOM; thus, the performance gap would become larger. However, when the number of UAVs is larger enough, NECTARs have achieved 100% *TSR* and the increase of UAV number would only promote the performance of RANDOM, so the performance gap would become smaller. Besides, NECTAR-M has achieved the highest *TSR* among all algorithms, and NECTARs perform better than RANDOM, which illustrates that our proposed mechanism and algorithms are robust.

From Figure 7(b), we can find that the *ETR* of NECTERs would slightly rise and, then, decline. This can be explained that the tasks arriving first need to respond first, and the number of candidate UAVs is too small, so the distance between the winner UAV and the task may not be short, so the time spent in transition may be too long, resulting in low efficiency. With the UAV number grows, the NECTARs would realize a higher *ETR*. As for the decline of the *ETR*, the reason is that when the UAVs are enough to undertake the arrived tasks, more UAVs would only bring resource idling, and thus, the *ETR* would decline. As for the direct decline of RANDOM's *ETR*, we attribute this to the random selection strategy. Because RANDOM has not taken the distance and time into comprehensive consideration such as NECTARs, the UAV resources are not sufficiently used. Thus, the *ETR* would decline.

We can conclude that the NECTARs would perform better than the baseline algorithm regardless of the number of UAVs. Consequently, the proposed mechanism and algorithms have good adaptability.

7. Conclusions and Future Work

In this study, we investigated the problem of dynamic real-time task assigning for the UAV swarm and proposed a novel distributed task assigning algorithm, NECTAR, based on agent intelligence and auction mechanism. The intelligent agent technology is employed to realize self-management for

UAVs. Also, the auction mechanism is used to complete the task assigning process in a distributed way. Our contributions include designing the task assigning models, and auction-based task assigning activities, as well as the collaboration processing stage and flows. In addition, we devised the calculation of bidding values on local devices to reduce communications. Besides, three selection strategies, i.e., the M strategy, D strategy, and P strategy, were proposed to determine the winner UAVs. Furthermore, we described NECTAR in detail by giving the algorithms for different agents. Finally, we conducted extensive simulation experiments to verify the effectiveness and efficiency, of which the results indicate that NECTAR is an effective task allocation algorithm for assigning dynamic tasks in the UAV swarm, and it is adaptable and robust when the parameters change.

Our NECTAR is the first of its kind found in the literature because NECTAR adopts intelligent agents and auction mechanism to address the dynamic tasks assigning problem in a distributed way for the UAV swarm, rather than allocate tasks in batch mode. In our future work, we would like to address the problems as follows. Firstly, we want to enhance NECTAR by taking weak communication conditions into consideration and study how they could complete the assigning process through some effective rules. Secondly, we plan to apply NECTAR into real UAV systems to accomplish task allocation.

Data Availability

The experiment data can be obtained by contacting the corresponding author (wdbao@nudt.edu.cn).

Conflicts of Interest

The authors declare that there are no conflicts of interest about the publication of this paper.

Authors' Contributions

(a) Chao Chen was involved in comprehensive processing of thesis writing, experiment, and data. (b) Weidong Bao was involved in idea guidance, paper guidance and revision, rewriting the Abstract and Introduction, and revision of the experiment code. (c) Tong Men conducted the experiment. (d) Xiaomin Zhu was involved in algorithm design guidance, pseudocode writing, and polishing the writing of the whole paper. (e) Ji Wang sorted out the related literature and helped Chao Chen in experimental analysis. (f) Rui Wang redrew the data graphs and corrected the syntax errors of the resubmitted version.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61872378 and 61773390), the Science Fund for Distinguished Young Scholars in Hunan Province (No. 2018JJ1032), and the Scientific Research Project of National University of Defense Technology (No. ZK19-03).

References

[1] T. Huang, D. Huang, Z. Wang, and A. Shah, "Robust tracking control of a quadrotor UAV based on adaptive sliding mode controller," *Complexity*, vol. 23, 2019.

[2] F. Sun, X. Wang, and Z. Rui, "Task scheduling system for UAV operations in agricultural plant protection environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 23, 2020.

[3] H. Meng and Y. Guo, "Automatic safety routing inspection of the electric circuits based on UAV light detection and ranging," *DEStech Transactions on Engineering and Technology Research*, vol. 23, 2017.

[4] L. Zhou, H. Ma, Z. Yang, S. Zhou, and W. Zhang, "Path loss modeling and evaluation: unmanned aerial vehicle communications," *IEEE Vehicular Technology Magazine*, vol. 34, 2020.

[5] M. Darrah, W. Niland, and B. Stolarik, "Multiple UAV dynamic task allocation using mixed integer linear programming in a SEAD mission," *Infotech*, vol. 34, 2006.

[6] E. Kendall, R. Phillip, and P. Meir, "Dynamic network flow optimization model for air vehicle resource allocation," in *Proceedings of the American Control Conference*, pp. 1853–1858, New York, NY, USA, 2001.

[7] D. Zhou, A. Zhang, and P. Yang, "Fixed-time output feedback consensus of second-order multi-agent systems with settling time estimation," *International Journal of Control Automation and Systems*, vol. 45, 2020.

[8] W. Zhu, L. Li, L. Teng, and Y. Wen, "Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding," *Chinese Journal of Aeronautics*, vol. 31, no. 2, pp. 339–350, 2018.

[9] Z. Zhu, B. Tang, and J. Yuan, "Multi-robot task allocation based on an improved particle swarm optimization approach," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, pp. 1–22, 2017.

[10] H. Wu, H. Li, R. Xiao, and J. Liu, "Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm," *Physica A: Statistical Mechanics and Its Applications*, vol. 491, pp. 127–141, 2018.

[11] C. Shumacher, P. Chandler, M. Pachter, and L. Pachter, "UAV task assignment with timing constraints via mixed-integer linear programming," in *Proceedings of the AIAA 3rd Unmanned Unlimited Systems Technical Conference*, p. 6410, New York, NY, USA, 2004.

[12] H. Wu and F. Zhang, "Wolf pack algorithm for unconstrained global optimization," *Mathematical Problems in Engineering*, vol. 91, 2014.

[13] S. Rasmussen, T. Shima, J. Mitchell, A. Sparks, and P. Chandler, "State-space search for improved autonomous UAVs assignment algorithm," in *Proceedings of the Conference on Decision and Control*, New York, NY, USA, 2004.

[14] A. Mehdi and P. Jonathan, "Decentralized task assignment for unmanned aerial vehicles," in *Proceedings of the European Control Conference Cdc-ecc 05 IEEE Conference on Decision & Control*, New York, NY, USA, 2005.

[15] A. Whitten, L. Choi, L. Johnson, and P. How, "Decentralized task allocation with coupled constraints in complex missions," in *Proceedings of the American Control Conference*, New York, NY, USA, 2011.

[16] X. Fu, J. Pan, X. Gao, B. Li, and K. Zhang, "Task allocation method for multi-UAV teams with limited communication bandwidth," in *Proceedings of the 2018 15th International*

Conference on Control, Automation, Robotics and Vision (ICARCV), New York, NY, USA, 2018.

[17] J. Yang, X. You, G. Wu, M. M. Hassan, A. Almogren, and J. Guna, "Application of reinforcement learning in UAV cluster task scheduling," *Future Generation Computer Systems*, vol. 95, no. 3, pp. 140–148, 2019.

[18] P. Sujit and R. Beard, "Distributed sequential auctions for multiple UAV task allocation," in *Proceedings of the American Control Conference IEEE*, New York, NY, USA, 2007.

[19] S. Ramchurn, J. Fischer, Y. Ikuno, F. Wu, J. Flann, and A. Waldock, "A study of human-agent collaboration for multi-UAV task allocation in dynamic environments," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, New York, NY, USA, 2015.

[20] K.-S. Kim, H.-Y. Kim, and H.-L. Choi, "A bid-based grouping method for communication-efficient decentralized multi-UAV task allocation," *International Journal of Aeronautical and Space Sciences*, vol. 21, no. 1, pp. 290–302, 2020.

[21] H.-X. Chen, Y. Nan, and Y. Yang, "Multi-UAV reconnaissance task assignment for heterogeneous targets based on modified symbiotic organisms search algorithm," *Sensors*, vol. 19, no. 3, p. 734, 2019.

[22] Z. Zhen, D. Xing, and C. Gao, "Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm," *Aerospace Science and Technology*, vol. 76, pp. 402–411, 2018.

[23] G. Fernandez, N. Kerle, and M. Gerke, "UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning," *Natural Hazards and Earth System Sciences*, vol. 15, no. 6, pp. 1087–1101, 2015.

[24] S. Adhau, M. L. Mittal, and A. Mittal, "A multi-agent system for distributed multi-project scheduling: an auction-based negotiation approach," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1738–1751, 2012.

[25] P. Ezhilchelvan and G. Morgan, "A dependable distributed auction system: architecture and an implementation framework," in *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems*, pp. 3–10, New York, NY, USA, 2001.

[26] UAV, "The 10 longest range unmanned aerial vehicles (UAVs)," 2020.

Hindawi
Complexity
Volume 2020, Article ID 8871086, 11 pages
https://doi.org/10.1155/2020/8871086

Research Article

Wireless Energy Transmission Link Optimization considering Microwave Energy Relay

Yuanming Song,[1,2] Yajie Liu ,[2] and Xu Yang[1,2]

[1]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China*
[2]*Hunan Key Laboratory of Multi-energy System Intelligent Interconnection Technology (HKL-MSI2T), Changsha 410073, Hunan, China*

Correspondence should be addressed to Yajie Liu; liuyajie@nudt.edu.cn

Received 27 August 2020; Accepted 8 September 2020; Published 16 September 2020

Academic Editor: Shi Cheng

With the widespread application of intelligent unmanned units and the development of wireless energy transmission technology, it has become an urgent requirement in the field of using wireless power transmission to extend the continuous working time of unmanned units. To improve the wireless transmission energy efficiency in practical application as much as possible based on the current development level of wireless transmission technology, this paper first enumerates and analyzes existing wireless energy supply methods and shows that microwave wireless energy transmission technology is most suitable in this study. Then, the necessity of power transmission relay in long-distance microwave wireless power transmission is discussed. Based on the idea of microwave energy relay transmission, the paper constructs a microwave wireless transmission link planning model which considers power relay and uses the evolutionary algorithm to solve the wireless transmission link planning model with two-layer optimization. Finally, the results show that, when the wireless transmission distance exceeds a certain threshold, adding relay nodes to the link can improve the energy transmission efficiency from the power supply node to the power receiving node.

1. Introduction

As a new type of power transmission technology, wireless energy transmission can be used to provide continuous energy supply in the working process of the unmanned units to extend their working time and working range [1, 2]. Roughly speaking, wireless energy transmission technology can be divided into near-field (nonradiation) and far-field (radiation) wireless energy transmission technology [3]. Subdivided from the principle of energy transfer, near-field wireless energy transmission can be divided into electric field coupling (electromagnetic induction) and magnetic field coupling (electromagnetic resonance) wireless energy transmission technologies [3, 4]; far-field wireless energy transmission includes ultrasonic wireless energy transmission, microwave wireless energy transmission, and laser wireless energy transmission [5]. These wireless energy transmission methods are different in energy transmission principle, effective energy transmission distance, transmission efficiency, and maturity. Among them, the more mature wireless energy transmission methods that can be initially put into the practical application are electric field coupling [6, 7], magnetic field coupling [8–10], and microwave wireless energy transmission [11, 12].

Because the unmanned unit tends to move in a larger range when performing tasks, the distance between the control terminal and the unmanned unit is as short as hundreds of meters, as long as several kilometers or even tens of kilometers. In this case, various near-field wireless energy transmission methods [13–15] that generally can only perform wireless energy transmission of the order of 10 m cannot meet the requirements of providing continuous and reliable energy supply for unmanned units at work. Compared with the electric field and magnetic field coupling wireless energy transmission method, microwave wireless energy transmission is also mature and its energy

transmission distance can reach several kilometers [11], which can well meet the needs of wireless energy transmission link for unmanned units at work [16, 17].

In a uniform atmosphere, the microwave beam attenuates exponentially [18]. In the processof microwave wireless energy transmission, if only the attenuation caused by the atmosphere, the microwave excitation efficiency, and the efficiency of microwave-electric energy conversion are considered, the microwave wireless energy transmission relay will increase the loss during excitation and reception. In this case, theenergy transmission efficiency of the point-to-point direct microwave energytransmission from the power supply end to the power receiving end must behigher than the efficiency of the power supply end forwarding microwave energy to the power receiving end through the relay end.

However, the loss of the microwave beam in the atmosphere is not only the exponential attenuation through the atmosphere but also the loss caused by the beam spreading in space during long-distance transmission (geometric attenuation) [19]. To minimize the latter loss, microwave wireless energy transmission generally uses a directional beam with a small excitation divergence angle for point-to-point energy transmission and at the same time modulates the beam so that most of its energy is concentrated in the center. Even if the above method is used to reduce the geometric attenuation since the excitation divergence angle cannot be zero, the directional beam cannot be regarded as a section of the cylinder from the microwave excitation node plane to the receiving node plane during long-distance transmission but should be regarded as a cone with the microwave excitation point as the apex and the microwave excitation divergence angle as the apex angle [20, 21]; the receiving node can only receive the energy distributed on the effective area of the receiving node antenna (antenna aperture) in the cross section of the beam cone [18]. Due to the limited receiving area of the receiving node antenna, when the geometric attenuation ratio of the point-to-point direct transmission is much greater than the re-excited relay efficiency after the relay node receives and conversion, it is very necessary to carry out the relay energy transmission in the microwave wireless energy transmission process. Figure 1 is a comparison diagram of geometric attenuation of energy transmission through two relays and without relay when the microwave excitation angle is the same.

Adding relay nodes to the link can reduce the geometric attenuation in microwave wireless energy transmission and improve the efficiency of wireless energy transmission from the power supply node to the power receiving node, but it will inevitably bring additional costs because of the purchase and operation of the relay nodes. Therefore, based on the distance between the power supply node to the power receiving node, the microwave excitation divergence angle of the power supply node, and the receiving antenna aperture of the power receiving node, the number and location of the relay node in the link and the required energy-receiving antenna aperture of relay node are needed to be optimized to achieve higher link wireless energy transmission efficiency with the lowest relay cost. In the decision-making process of link construction, it is necessary to increase the number of relays within a certain range and increase the aperture of the energy-receiving antenna of the relay node to improve the efficiency of link wireless energy transmission; this will lead to an increase in the cost of the link. In the construction, the requirement of minimizing the relay cost and the requirement of maximizing the wireless energy transmission efficiency of the link restricts each other, which constitutes a typical dual-objective optimization problem.

This paper is based on the microwave wireless energy transmission technology to construct the wireless energy transmission link considering the energy transmission relay. According to the actual situation of the power supply node and the power receiving node, the relay nodes are added to the link to improve the overall link energy transmission efficiency. Based on the multiobjective evolutionary algorithm, the number and location of the relay nodes in the link and the microwave energy-receiving antenna aperture of the relay nodes are optimized. The main contributions of this paper are as follows:

(1) This paper analyzed the principles, advantages, and disadvantages of current wireless energy transmission methods, explained that microwave wireless energy transmission is the most suitable method for continuous energy supply to unmanned units at work, and discussed the necessity of microwave energy transmission relay.

(2) Unit models for the power supply node, energy receiver node and relay node of the energy transmission link are established, and a microwave wireless energy transmission link planning model that considers energy transmission relay based on the unit model is constructed in this study.

(3) The evolutionary algorithm (NSGA-II) is used to solve the planning model of the microwave wireless energy transmission link considering the energy transmission relay based on a two-layer iterative optimization method. The solution result verifies the effectiveness of the microwave energy relay transmission proposed in this paper.

2. Literature Review

The current research on wireless energy transmission links is mostly based on magnetic field coupling wireless energy transmission technology. The team of Professor Sun from Chongqing University proposed the concept of wireless power transmission network based on magnetic field coupling wireless energy transmission technology [22]. In this wireless power transmission network, multiple relay terminals use their resonant coils to form a resonance link and use the strong coupling between the resonators to form a "high-efficiency energy transmission channel." The relay process does not involve the conversion of energy forms. Using this principle, in literature [23], the transmitter transmits power to the receiver 2 meters away through multiple relay nodes, which improves the effective distance

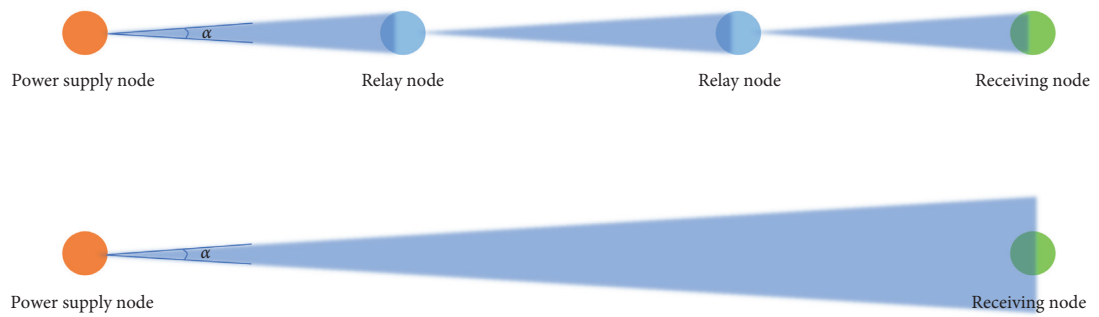


Figure 1: Comparison diagram of geometric attenuation.

of wireless energy transmission. Literature [24] assumes that there are multiple relay nodes randomly distributed in a certain area and uses the ant colony algorithm to realize the optimal selection of relay nodes in the energy transmission link. Under the same assumption, literature [25] uses the cross-entropy algorithm to optimize the energy transmission link, which avoids the problem of ant colony algorithm that is easy to fall into the local optimum in path optimization. Based on whether the wireless power transmission network has a certain transmitting node and whether the transmitting node can provide unlimited power, the wireless power transmission network defined in literature [22] is further divided into active injection wireless power transmission network and passive injection wireless power transmission network in literature [26], and it is proved through experiments that the cellular genetic algorithm can effectively solve the path optimization of wireless power transmission network with active injection and passive injection.

The related research of microwave wireless energy transmission is almost entirely focused on improving the efficiency of DC-microwave conversion, receiving rectification and DC synthesis in the process of direct energy transmission between two points [11, 27]. There is no related research on microwave wireless energy relay transmission. Starting from the microwave wireless energy transmission experiment reported by NASA in 1974, which achieved a DC-to-DC conversion efficiency of 6.7% at a distance of 1.54 km [28], researchers have been committed to improving the overall transmission efficiency of microwave wireless energy transmission. In 2009, the C-band microwave wireless energy transmission system researched by Sichuan University achieved 70% rectification efficiency under the conditions of working frequency 5.8 GHz and transmitting power 600 W; in 2012, a research team from Kyoto University in Japan built an S-band microwave wireless energy transmission system using phased array magnetron; the wireless energy transmission system has a microwave-to-DC conversion efficiency of 54% when the transmission power is 1.9 kW [11]. Literature [27] designed a high-performance microwave source for the power supply node of a microwave wireless power transmission system, using phase-locked loop-based frequency synthesis technology and a monolithically integrated power chip to simplify the circuit system structure.

The idea of relay energy transmission in the wireless power transmission network based on magnetic field coupling wireless energy transmission technology points out a good direction for wireless energy link transmission.

However, limited by the technical principle of magnetic field coupling wireless energy transmission, the wireless power transmission network based on magnetic field coupling wireless energy transmission technology can only provide radio for energy-receiving devices within a range of tens of meters even after multiple relays [23, 26], which is difficult to meet the energy demand of the unmanned unit in this paper. Considering that the microwave wireless energy transmission technology is relatively mature [11], it can carry out the wireless energy transmission of the kilometer level with higher efficiency, and this paper is based on the microwave wireless energy transmission technology to construct the wireless energy transmission link for the working unmanned unit. The position and quantity of the relay node required by the link and the aperture of the energetic antenna at the relay node are optimized to minimize the geometric attenuation of microwave energy in wireless energy transmission and improve the overall efficiency of energy transmission.

What needs to be clear is that, compared with the magnetic field coupling relay energy transmission of Sun Yue's team, the microwave energy relay transmission proposed in this paper uses a similar form of wireless energy link transmission, but whether the energy transmission principle or the meaning of adding relay nodes is completely different from it.

3. Model Construction

3.1. Unit Model. In this paper, the microwave wireless energy transmission link that considers energy relay transmission consists of a fixed energy supply platform (power supply node) as a wireless power transmission source, a relay platform (relay node) that functions as an energy relay in the link, and an unmanned unit (power receiving node) that receives energy. To simplify the calculation, this paper assumes that the microwave beam used in wireless energy transmission is a plane wave and the energy density on the beam section obeys a uniform distribution [20, 21]; the maximum microwave transmission power of the power supply node and the maximum receiving and forwarding

power of microwave energy at the relay node is determined according to the power requirements and the microwave energy-receiving capability of the receiving node; this study does not involve the optimization of the transmitting and receiving power of each node in the link.

3.1.1. Model of the Power Supply Node. The main parameters of the power supply node (fixed energy supply platform) include the microwave beam excitation angle α_{wav}^{ps}, the conversion efficiency (η_{wes}^{ps}) of converting electrical energy into microwave energy, and the three-dimensional coordinates (POS_{ps}) of the microwave source. When the microwave source at the power supply node emits microwaves in a certain direction in space, the energy distributed on the beam cross section ($P_{mwsec}(L_{wet})$) with a distance of L_{wet} from the power supply node can be calculated by the following formula:

$$P_{mwsec}(L_{wet}) = \eta_{wes}^{ps} \cdot \kappa \cdot e^{-\gamma \cdot L_{wet}} \cdot P_{wes}^{ps}, \tag{1}$$

where P_{wes}^{ps} is the actual microwave transmit power at the power supply node; γ is the atmospheric attenuation coefficient, which is related to the atmospheric conditions on the beam transmission path; and κ is the tracking loss coefficient. The values of γ and κ are determined based on actual weather conditions and equipment conditions during transmission.

3.1.2. Model of the Power Receiving Node. The unmanned unit as the power receiving node receives the microwave energy directly transmitted by the power supply node or forwarded by the relay node during the process of completing the predetermined work.

The parameters of power receiving node include the receiving antenna aperture S_{wer}^{pr}, the efficiency (η_{wer}^{pr}) of converting the microwave energy distributed on the effective area of the receiving antenna into electrical energy, and the three-dimensional coordinates of the equivalent center point of the receiving antenna (using this as the power receiving node coordinate POS_{pr}). When the distance between the power receiving node and the microwave emitting source is L_{wet}, the electric power P_{wer}^{pr} converted by microwave energy is calculated by the following formula:

$$P_{wer}^{pr} = \eta_{wer}^{pr} \cdot \min\left(\frac{S_{wer}^{pr}}{S_{wav}(\alpha_{wav}, L_{wet})}, 1\right) \cdot P_{mwsec}(L_{wet}). \tag{2}$$

Under the plane wave assumption, the beam cross-sectional area $S_{wav}(\alpha_{wav}, L_{wet})$ in the formula is related to the beam cone angle α_{wav} and the transmission distance L_{wet}:

$$S_{wav}(\alpha_{wav}, L_{wet}) = \pi\left(\tan\left(\frac{\alpha_{wav}}{2}\right) \cdot L_{wet}\right)^2, \quad \alpha_{wav} \in (0, \pi). \tag{3}$$

3.1.3. Model of the Relay Node. The relay node plays the role of energy relay and forwarding in the link. Its main parameters include the aperture of the receiving antenna S_{wer}^{re}, the efficiency (η_{wer}^{re}) of converting the microwave energy distributed on the receiving antenna into electric energy, the excitation angle α_{wav}^{re} of the microwave beam, the conversion efficiency (η_{wes}^{re}) of converting electric energy into microwave energy, and the three-dimensional coordinates POS_{re} of the relay node.

When the distance between the relay node and the microwave emission source is L_{wet}, the electric power P_{wer}^{re} converted by microwave energy is calculated by the following formula:

$$P_{wer}^{re} = \eta_{wer}^{re} \cdot \min\left(\frac{S_{wer}^{re}}{S_{wav}(\alpha_{wav}, L_{wet})}, 1\right) \cdot P_{mwsec}(L_{wet}). \tag{4}$$

While receiving energy, the relay node begins to forward microwave energy to the power receiving node or another relay node, and the loss of the transformer and stabilizing circuit in the process is merged into the conversion efficiency (η_{wes}^{re}) of the electric energy to microwave energy. When the relay node emits microwaves, the energy distributed on the beam section at a distance of L_{wet} from it can be calculated by the following formula:

$$P_{mwsec}(L_{wet}) = \eta_{wes}^{re} \cdot \kappa \cdot e^{-\gamma \cdot L_{wet}} \cdot P_{wer}^{re}. \tag{5}$$

3.2. Modeling of Microwave Wireless Energy Transmission Link. The relay node in the wireless energy transmission link can have various forms according to actual needs, such as a power relay drone, a relay powered vehicle, or a fixed relay platform. Figure 2 is a schematic diagram of a microwave wireless energy transmission link with power relay drones as relay nodes; the use of power relay drones in the wireless energy transmission link can avoid the interference from ground obstacles. In this paper, the planning model chooses power relay drones as relay nodes to avoid interference from ground obstacles.

Link transmission efficiency is the most important attribute of a wireless energy transmission link. The goal of relaying energy transmission is to reduce the geometric attenuation of microwave energy during transmission. For any two nodes (i, j) in the link, the energy transfer efficiency from node i to node j is calculated by the following formula:

$$\eta_{i-j} = \kappa_{i-j} \cdot \gamma^{L_{i-j}} \cdot \eta_{wes}^{i} \cdot \eta_{wer}^{j} \cdot \min\left(\frac{S_{wer}^{j}}{S_{wav}^{i}}, 1\right). \tag{6}$$

The cross-sectional area S_{wav}^{i} of the microwave beam emitted by node i at node j is calculated by the following formula:

$$S_{wav}^{i}\left(\alpha_{wav}^{i}, L_{i-j}\right) = \pi \cdot \left(\tan\left(\frac{\alpha_{wav}^{j}}{2}\right) \cdot L_{i-j}\right)^2, \tag{7}$$

where L_{i-j} is the straight-line distance from node i to node j.

3.3. Optimization Objectives. The primary goal of wireless energy transmission link construction is to maximize the

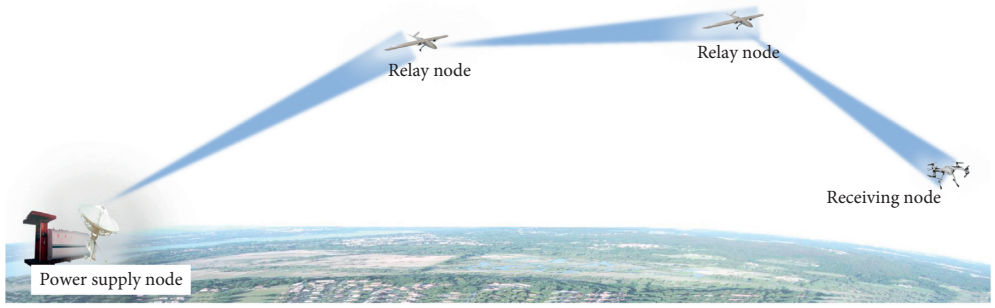


Figure 2: Schematic diagram of microwave wireless energy transmission link with multiple relays.

energy transmission efficiency during the energy transmission process from the power supply node to the power receiving node, and the additional cost of adding a relay node must also be considered. Therefore, maximizing link energy transfer efficiency and minimizing relay cost of the link are the optimization objectives for constructing wireless energy transmission links.

3.3.1. Energy Transfer Efficiency of the Link. The energy relay transmission in the microwave wireless energy transmission link is to reduce geometric attenuation and improve the energy transmission efficiency of the link from the power supply node to the power receiving node.

When there are n relay nodes in the link, the link transmission efficiency at a certain moment is calculated as follows:

$$\eta_{\text{link}} = \eta_{\text{ps-re}}\left(L_{\text{ps-re}}\right) \cdot \prod_{i=1}^{n-1} \eta^{i}_{\text{re-re}}\left(L^{i}_{\text{re-re}}\right) \cdot \eta_{\text{re-pr}}\left(L_{\text{re-pr}}\right), \tag{8}$$

where $\eta_{\text{ps-re}}(L_{\text{ps-re}})$, $\eta^{i}_{\text{re-re}}(L^{i}_{\text{re-re}})$, and $\eta_{\text{re-pr}}(L_{\text{re-pr}})$ are the transmission efficiency functions from the power supply node to the nearest relay node, the i^{th} relay node to the next relay node, and the last relay node to the power receiving node.

The specific calculation method of the transmission efficiency at energy time refers to formulae (7) and (8); A, B, and C are, respectively, the power supply node to the nearest relay node, the i^{th} relay node to the next relay node, and the microwave energy transmission distance from the last relay node to the power receiving node. The specific calculation method of the transmission efficiency when energy is directly transmitted between two points refers to equations (7) and (8). $L_{\text{ps-re}}$, $L^{i}_{\text{re-re}}$, and $L_{\text{re-pr}}$ are the microwave transmission distance from power supply node to the nearest relay node, the i^{th} relay node to the next relay node, and the last relay node to the power receiving node, respectively.

3.3.2. Relay Cost of the Link. In energy-related network planning, the cost of a network unit generally includes three parts: acquisition cost, maintenance cost, and operating cost [29]. Since the planning model in this paper is aimed at verifying the effectiveness of relay energy transmission in the link, only the purchase cost of the relay nodes is considered in the optimization. When there are n relay nodes in the link, for the relay cost C_{link}, there is

$$C_{\text{link}} = \sum_{i=1}^{n}\left(C^{i}_{\text{fmre}} + C^{i}_{\text{wer}}\left(S^{i}_{\text{wer}}\right)\right), \tag{9}$$

where C^{i}_{fmre} is the purchase cost of the microwave energy-receiving equipment at the relay node and $C^{i}_{\text{wer}}(S^{i}_{\text{wer}})$ is the purchase cost of the microwave energy-receiving equipment related to the energized antenna aperture S^{i}_{wer} of the i^{th} relay node.

4. Model Solving

In the microwave wireless energy transmission link planning, it is necessary to consider the two mutually restrictive objectives of maximizing the energy transmission efficiency of the link and minimizing the link relay cost, which is a typical dual-objective optimization problem. Among the various existing multiobjective evolutionary algorithms [30, 31], the fast nondominated sorting genetic algorithm with elite strategy (NSGA-II) [32], as an excellent algorithm, has been widely used in many disciplines [33, 34]; its characteristics also fit for the experiments in this study. Therefore, this paper uses NSGA-II to solve the wireless energy transmission link planning model considering energy relay transmission.

To achieve the continuous energy supply of the unmanned unit at work with the relay energy transmission link based on microwave wireless energy transmission technology, while maximizing the transmission efficiency, the number and location of the relay nodes in the link and the aperture of the relay node antenna need to be optimized based on the coordinates and the performance (beam excitation angle, energy-receiving antenna aperture, etc.) of the power supply node and the power receiving node.

Since the number of relay nodes in the decision variable determines the number of coordinates of the relay nodes, encoding the number and the coordinates of relay nodes in the same individual will cause the chromosome length of each individual to be different. In evolutionary calculations, chromosomes of different lengths cannot be crossed. Therefore, the inner layer optimization of the positions of the relay nodes is performed separately, and the optimization of the number and the antenna aperture of relay nodes

are completed based on the returned optimization results and other relevant parameters.

Figure 3 shows the solution process of the wireless energy transmission link planning model considering energy relay transmission: the solution adopts the double-layer optimization method. The decision variables of the outer layer optimization are the number and the aperture of the energy-receiving antenna of relay nodes. The optimization objectives are to maximize the link energy transmission efficiency and minimize the link relay cost; the decision variable of the inner optimization is the relay node coordinates, and the optimization objective is to maximize the link energy transmission efficiency. In solution, first, enter the number of population individuals and the number of optimization goals, set the value range of the decision variables, generate the outer optimized initial population, and set the maximum evolutionary generation. Before calculating the objective function value, based on the number and the antenna aperture of relays in the individual, the inner layer optimization calculation is performed on the position of each relay in the link with the objective of maximizing the energy transmission efficiency of the link. Then, save the coordinates of each relay node in the obtained optimization results and use the link energy transfer efficiency of the inner optimization result as one of the objective values of the current individual of the outer layer optimization; the other objective value of the outer layer optimization (relay cost of the link) is calculated based on the input parameters and related formulas. After fast nondominant sorting, crowding calculation, selection, crossover, mutation, and other operations on the initial population, a new generation of population is obtained and double-layer loop iterations are continuously performed before reaching the maximum number of evolutionary iteration. When reaching the maximum number of evolutionary iteration, terminate the loop and output the solution result.

Based on the optimization objectives, decision variables and constraints that have been determined previously, the dual-objective planning problem of wireless energy transmission links considering energy relay transmission can be described as follows:

(a) Outer layer optimization

Objective function:

$$\min F_{\text{obj}} = \left(1 - \eta_{\text{link}}, C_{\text{link}}\right). \tag{10}$$

Restrictions:

$$N_{\text{re}} \geq 0. \tag{11}$$

$$S^{\text{re}}_{\text{wermin}} \leq S^{\text{re}}_{\text{wer}} \leq S^{\text{re}}_{\text{wermax}}. \tag{12}$$

(b) Inner layer optimization

Objective function:

$$\min\left(1 - \eta_{\text{link}}\right). \tag{13}$$

Restrictions:

$$X_{\text{ps}} \leq X_{\text{re}} \leq X_{\text{pr}}. \tag{14}$$

$$Y_{\text{ps}} \leq Y_{\text{re}} \leq Y_{\text{pr}}. \tag{15}$$

$$Z_{\text{ps}} \leq Z_{\text{re}} \leq Z_{\text{pr}}. \tag{16}$$

In the previous formula, F_{obj} represents the objective function including two objectives. The decision variables N_{re} and $S^{\text{re}}_{\text{wer}}$ are the number of relay nodes in the link and the aperture of the energy-receiving antenna at the relay nodes. $S^{\text{re}}_{\text{wermax}}$ and $S^{\text{re}}_{\text{wermin}}$ are the upper and lower limits of the aperture of the energy-receiving antenna at the relay nodes; $(X_{\text{ps}}, Y_{\text{ps}}, Z_{\text{ps}})$ and $(X_{\text{pr}}, Y_{\text{pr}}, Z_{\text{pr}})$ are the coordinates of the power supply node and the energy-receiving node, respectively. The coordinate $(X_{\text{re}}, Y_{\text{re}}, Z_{\text{re}})$ of the relay node is the decision variable for inner optimization, and its value range must meet formulae (13)–(15).

When using NSGA-II to solve the dual-objective planning problem of wireless energy transmission link by two-layer iteration, set the number of individuals in the outer optimization population to 50 and the maximum number of iterations to 100; set the number of individuals in the inner optimization population to 25, the maximum number of iterations is floating according to the actual number of decision-making variables in the inner optimization; the genetic operators all use simulated binary crossover (SBX) and polynomial mutation (PM); for specific parameters, refer to related discussions in [32, 35, 36] and the actual situation in algorithm debugging. The coding of the chromosome in the solution adopts real number coding. In the outer layer optimization, each chromosome is composed of 2 genes (decision variables): $[N_{\text{re}} \mid S^{\text{re}}_{\text{wer}}]$, and in the inner optimization, each chromosome is composed of n genes $(n = 3 \times N_{\text{re}})$: $[X^1_{\text{re}} \mid Y^1_{\text{re}} \mid Z^1_{\text{re}} \mid X^2_{\text{re}} \mid Y^2_{\text{re}} \mid Z^2_{\text{re}} \mid \ldots \mid X^n_{\text{re}} \mid Y^n_{\text{re}} \mid Z^n_{\text{re}}]$.

5. Case Study

5.1. Experimental Data. To verify the effectiveness of energy relay transmission in microwave wireless energy transmission, wireless energy transmission link optimization is performed based on the equipment technical parameters and cost data sorted out from related literature [11, 12, 18, 21]. The actual form of the relay node is an unmanned aerial vehicle carrying energy relay transmission equipment. The optimization process does not consider the occlusion caused by terrain changes or other reasons and assumes that the atmospheric environment in the experiment remains uniform and undisturbed.

Ten times of link planning was carried out in the experiment, and the relative distance between the power supply node and the power receiving node increased linearly in each time. The coordinate system is established with the

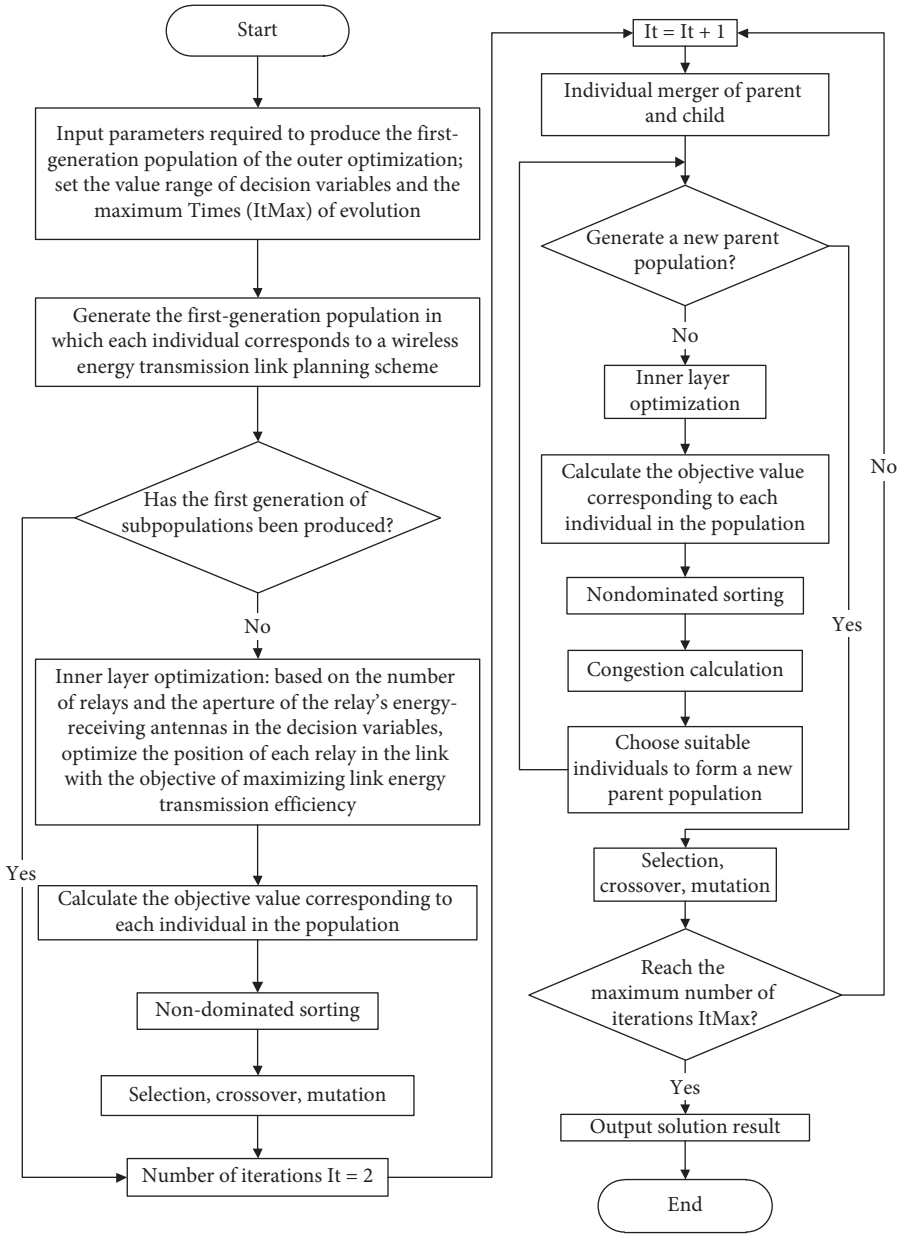


Figure 3: The solution process of the wireless energy transmission link planning model considering energy relay transmission.

power supply node as the origin. The X-Y plane position of the power receiving node in the 10 times of planning is shown in Figure 4. The related parameter settings required in the planning are shown in Tables 1 and 2.

5.2. Experimental Results. When using the CCMO and NSGA-II algorithms to solve the problem of wireless energy transmission link planning by a two-layer iterative method, firstly generate an initial population containing 50 individuals of outer optimization and then perform 100 times of two-layer iterative evolution. During iteration, each individual in each generation of the outer layer optimization must complete an inner layer optimization to calculate the value of objectives. The optimal Pareto fronts obtained by solving 10 times in the experiment are displayed in colors on the same graph, as shown in Figure 5. In the figure, the ordinate "Objective 1" is $1 - \eta_{\text{link}}$, and the abscissa "target value 2" is C_{link}.

It can be seen from Figure 5 that there is an obvious negative correlation between the relay cost (C_{link}) and energy transmission loss rate ($1 - \eta_{\text{link}}$) of the link in the solution results of the 10 times planning. To reduce the energy transmission loss rate (that is, to improve the energy transmission efficiency η_{link}) of the link will increase the link relay cost. In other words, a wireless energy transmission link with higher energy transmission efficiency requires a higher link relay cost; the optimal link planning scheme will be obtained by weighing these two optimization objectives. No planning scheme can minimize the relay cost and the energy loss rate of the link at the same time.

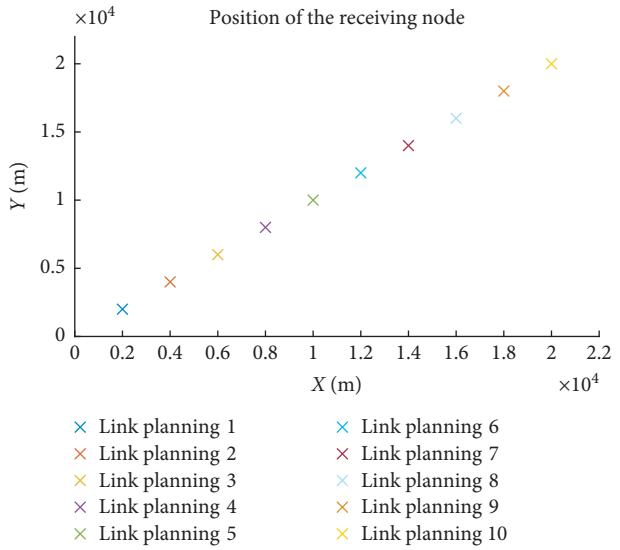


Figure 4: The position of the receiving node in the 10 times of optimization.

Table 1: Experimental equipment parameters.

η_{wes}^{ps}	0.80
η_{wes}^{re}	0.90
η_{wer}^{pr}	0.95
η_{wer}^{re}	0.95
α_{wav}^{ps} (rad)	9.81×10^{-4}
α_{wav}^{re} (rad)	9.81×10^{-4}
γ	1.01×10^{-3}
κ	0.95
S_{wer}^{pr} (m^2)	1
Z_{ps} (m)	0
Z_{pr} (m)	500
S_{wermax}^{re} (m^2)	30
S_{wermin}^{re} (m^2)	0

In the optimal Pareto set obtained after 100 iterations, the number of nondominated candidate solutions is 50, that is, 50 nondominated dynamic wireless energy transmission link planning schemes are finally obtained (the same in Link Planning 1, 50 schemes have been obtained, but many schemes have similar target values and are not easy to distinguish on the graph). These schemes are all theoretically optimal schemes, and decision-makers can choose the most suitable planning scheme according to their preferences. In terms of scheme selection, many methods can be used to assist decision-makers in making decisions [37–39]. For example, when a decision-maker uses the expectation level method [40] to choose a solution, he can specify that a certain expectation value needs to be achieved on a certain objective and then choose the solution that can achieve this expectation value while performing the best on another objective.

Besides, Figure 5 shows that, when the distance between the power supply node and the power receiving node increases linearly, the maximum energy transmission efficiency that the link can achieve decreases exponentially. This is because the relay node added to the link can only reduce the geometric attenuation in the microwave wireless energy transmission but cannot reduce the atmospheric attenuation in the energy transmission process, so the highest energy transmission efficiency that the link can achieve is mainly affected by the transmission process. The influence of atmospheric attenuation is reflected in the linear increase of the energy transmission distance, and the maximum energy transmission efficiency decays exponentially.

Another feature presented in Figure 5 is that the Pareto front obtained by the solution shows a linear negative correlation when the link relay cost is small and shows a nonlinear negative correlation when the link relay cost is high.

The Pareto front obtained from the sixth solution is selected to further analyze this feature. The Pareto front obtained by the sixth solution is shown in Figure 6; the 10 schemes uniformly drawn from the Pareto front obtained by the sixth solution are shown in Table 3. The target value in the table has converted the energy transmission loss rate $(1 - \eta_{link})$ into the average link energy transmission efficiency η_{link} of the link, and the schemes in the table are arranged in ascending order according to the value of η_{link}.

It can be seen from Figure 6 that the objective values show a linear negative correlation in the x-axis coordinates (0, 290,000) and (420,000, 550,000). This is because the schemes add the same number of relays in the link (Table 3, schemes 2–4, 5, and 6), and the geometric attenuation during energy transmission decreases linearly with the increase of the antenna aperture mounted on the relay node.

The Pareto front has a gap between the intervals (0, 290,000) and (420,000, 550,000). This is because the scheme in the (420,000, 550,000) interval adds 1 relay node compared with the scheme in the (0, 290,000) interval, and the high single unit cost (29999 CNY) of the relay node causes this gap on the front. When the front edge is in (550,000, 830,000), the two objective values show a distinct nonlinear negative correlation. This is because there are many relay nodes in the link on this part of the front, and the geometric attenuation in the wireless energy transmission process is already very small; further increasing the aperture of the energy-receiving antenna of the relay node has a reduced effect on reducing geometric attenuation. The sudden change of the slope in the middle of this part of the front is because the number of relay nodes on both sides of the slope sudden change point has changed. In general, the shape of the Pareto front obtained by the solution is in line with theoretical expectations and the two-layer iterative solution has found the optimal Pareto front.

In Table 3, the energy transmission efficiency of the link is generally low due to the long distance between the power supply node and the power receiving node (12 kilometers). Scheme 1 does not add a relay node to the link. At this time, although the link relay cost is 0, the energy transmission efficiency of the link is only 0.057%. Wireless energy transmission with such a low energy transmission efficiency will result in huge energy waste, and this scheme is not feasible in practical applications. In contrast, scheme 4 only adds a repeater to the link (the energy-receiving antenna aperture is 19 m^2), which makes the energy transfer efficiency increasing

Table 2: Equipment cost parameters.

Relay node purchase cost (CNY)	29999
Cost of the energy-receiving antenna and related rectifier equipment (CNY/m^2)	10400

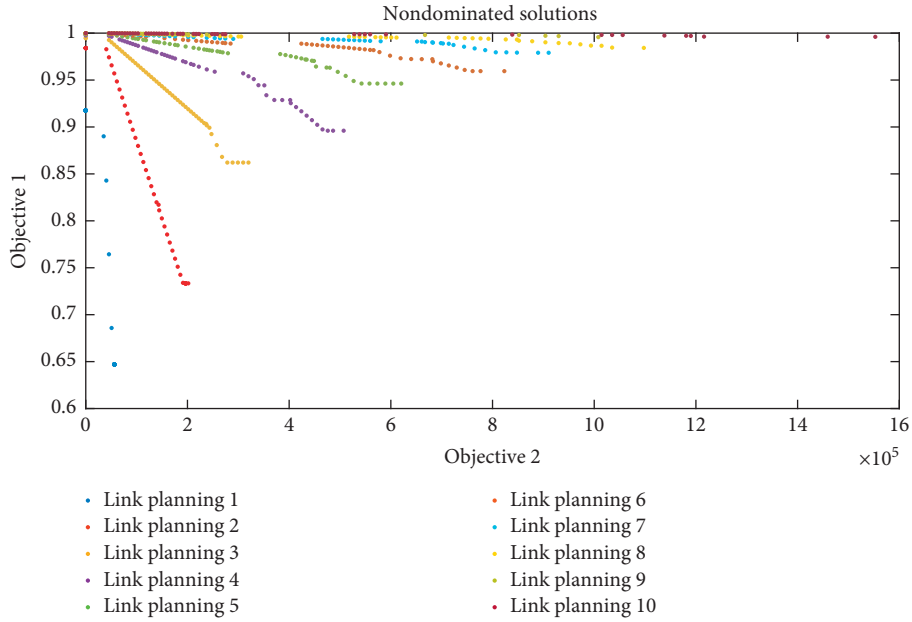


Figure 5: The optimal Pareto front obtained by solving 10 times.

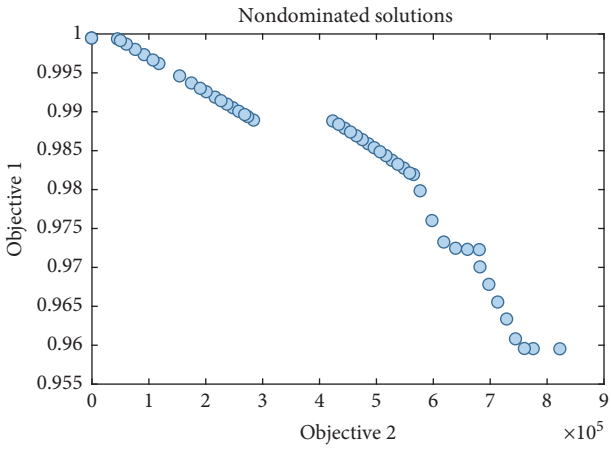


Figure 6: The Pareto front obtained by link planning 6.

Table 3: The Pareto solution of link planning 6.

Serial number	Decision variable		Objective value	
	N_{re}	S^{re}_{wer} (m^2)	η_{link} (%)	C_{link} (CNY)
1	0	0	0.057547	0
2	1	4.5	0.204328	76799
3	1	12	0.544805	154799
4	1	19	0.862794	227599
5	2	17.5	1.123708	423998
6	2	21.5	1.519537	507198
7	5	8	1.811696	565995
8	4	13.5	2.777558	681596
9	3	23.5	4.051886	823197

by about 15 times compared with scheme 1 (to 0.86%). Of course, the energy transfer efficiency of 0.86% also has lower practical feasibility, but the huge improvement of scheme 4 compared with scheme 1 fully shows that adding a suitable number of relay nodes to the link can effectively improve the average energy transmission efficiency of the link.

After the previous in-depth analysis of the optimal Pareto frontier obtained by the solution, the optimal Pareto solution is generally in line with the expected solution result based on the input data information and model constraints. The solution scheme shows a good optimization effect, and the effectiveness of the wireless energy transmission link multiobjective planning model in this paper has been verified; energy relay transmission in the microwave wireless energy transmission link in this paper can effectively improve the energy transmission efficiency of the link. Decision-makers can weigh the energy transfer efficiency requirements and economic cost budgets to choose a suitable scheme on the optimal Pareto frontier.

6. Conclusion

Aiming at the problem of realizing wireless power supply for unmanned units at work based on the current wireless energy transmission technology level and improving the energy transmission efficiency as much as possible, this paper proposes an energy supply method based on microwave wireless energy transmission technology, which transmits microwave energy from the power supply node to the power receiving node through a certain number of relay nodes.

First, the paper analyzes the characteristics of several types of currently more mature wireless energy transmission technologies and concludes that microwave wireless energy transmission is the most suitable way in the current application context. Then, under the condition that the receiving antenna aperture of the receiving node is fixed, the feasibility of effectively reducing the geometric attenuation in long-distance microwave wireless energy transmission by relaying microwave energy is discussed; the planning model of microwave wireless energy transmission link considering energy relay transmission is established and solved. Finally, an experiment was designed to verify the important role of microwave energy relay transmission in reducing the geometric attenuation in the process of microwave wireless energy transmission.

The geometric attenuation of microwave wireless energy transmission increases sharply with the increase of the energy transmission distance. When the divergence angle of the microwave excitation at the power supply node is constant and the antenna aperture of the power receiving node is limited, adding an appropriate number of relay nodes between the power supply node and the power receiving node of long-distance microwave wireless energy transmission can effectively reduce the geometric attenuation during transmission. When the number and the receiving antenna aperture of relay nodes are optimal, the effect of nearly completely avoiding geometric attenuation during transmission can be achieved at a lower additional cost, and the transmission efficiency from the power supply node to the receiving node can be maximized.

Besides, because the relay nodes can be deployed flexibly, the power supply node can also use microwave energy relay transmission to transmit energy to the power receiving node which is blocked by obstacles during point-to-point linear transmission. These characteristics reflect the feasibility of providing continuous energy supply to unmanned units in operation based on the current microwave wireless energy transmission technology, and the value of microwave energy relays transmission in practical applications.

The follow-up research will further enrich the planning model of microwave wireless energy transmission links based on the existing microwave wireless energy transmission technology and the idea of relay energy transmission. The research results of this paper can provide theoretical support for the subsequent theoretical research and engineering design of medium/long-distance wireless energy transmission links and wireless energy transmission networks.

Data Availability

The data used to support the study are presented within the article. Other data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Fitra and S. Elvy, "Wireless power for mobile battery charger," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 6, no. 2, pp. 278–285, 2017.

[2] S. Bi, Y. Zeng, and R. Zhang, "Wireless powered communication networks: an overview," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 10–18, 2016.

[3] Z. H. Zhao, *Research on Energy Transmission Mechanism and Optimization of Wireless Power Transmission Network*, Chongqing University, Chongqing, China, 2016, in Chinese.

[4] W. Ni, F. Ding, J. Zong, W. W. Ji, and X. J. Liu, "Research progress of electric energy wireless transmission and energy interconnection technology," *Power Technology*, vol. 43, no. 2, pp. 357–360, 2019.

[5] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[6] J. P. K. Sampath, D. M. Vilathgamuwa, and A. Alphones, "Efficiency enhancement for dynamic wireless power transfer system with segmented transmitter array," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 1, pp. 76–85, 2016.

[7] F. Y. Lin, G. A. Covic, and J. T. Boys, "Evaluation of magnetic pad sizes and topologies for electric vehicle charging," *IEEE Transactions on Power Electronics*, vol. 30, no. 11, pp. 6391–6407, 2015.

[8] D. Lin, C. Zhang, and S. Y. R. Hui, "Mathematic analysis of omnidirectional wireless power transfer-part-II three-dimensional systems," *IEEE Transactions on Power Electronics*, vol. 32, no. 1, pp. 613–624, 2017.

[9] J. Dai and D. C. Ludois, "Capacitive power transfer through a conformal bumper for electric vehicle charging," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 3, pp. 1015–1025, 2016.

[10] Y. Li, R. Mai, T. Lin, H. Sun, and Z. He, "A novel WPT system based on dual transmitters and dual receivers for high power applications: analysis, design and implementation," *Energies*, vol. 10, no. 2, p. 174, 2017.

[11] S. G. Li, Z. L. Chen, L. Song et al., "Research on ku-band microwave wireless energy transmission system technology," *Journal of Microwaves*, vol. 35, no. 4, pp. 56–61, 2019, in Chinese.

[12] H. H. Ma, H. Xu, X. Li et al., "A highly efficient microwave wireless power transmission system," *Space Electronic Technology*, vol. 1, pp. 1–5, 2016.

[13] R. Mai, L. Ma, Y. Liu, P. Yue, G. Cao, and Z. He, "A maximum efficiency point tracking control scheme based on different cross coupling of dual-receiver inductive power transfer system," *Energies*, vol. 10, no. 2, p. 217, 2017.

[14] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless charging technologies: fundamentals, standards, and network applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2016.

[15] W. Chen, C. Liu, C. Lee, and Z. Shan, "Cost-effectiveness comparison of coupler designs of wireless power transfer for electric vehicle dynamic charging," *Energies*, vol. 9, no. 11, p. 906, 2016.

[16] D. E. Becker, R. Chiang, C. C. Keys et al., "Photovoltaic-concentrator based power beaming for space elevator application," in *Proceedings of the AIP Conference*, pp. 271–281, Mumbai, India, May 2010.

[17] AUVSI, *Lockheed Demonstrate Real-World Laser Power*, Association for Unmanned Vehicle Systems International,

Chicago, IL, USA, 2012, https://www.flightglobal.com/news/articles/auvsi-lasermotive-lockheed-demonstrate-real-world-laser-375166/.

[18] M. Liu, X. G. Liu, J. Y. Mou, and X. J. Wu, "Analysis of optical power attenuation model for wireless optical communication," *Infrared and Laser Engineering*, vol. 41, no. 8, pp. 2136–2140, 2012, in Chinese.

[19] A. Prokes, "Atmospheric effects on availability of free space optics system," *Optical Engineering*, vol. 48, no. 6, 2009.

[20] Links (with focus on Malaysia). Proceeding of the international conference in computer and communication engineering, 2008: 13-15.

[21] M. S. Sheikh, P. Kohldorfer, and E. Leitgeb, "Channel modeling for terrestrial free space optical links," in *Proceedings of 2005 7th International Conference Transparent Optical Networks*, pp. 407–410, New York, NY, USA, 2005.

[22] Y. Sun, X. Dai, C. S. Tang et al., "Distributed wireless power transmission network," *Power Electronics*, vol. 3, pp. 59–61, 2010, in Chinese.

[23] C. K. Lee, W. X. Zhong, and S. Y. R. Hui, "Effects of magnetic coupling of nonadjacent resonators on wireless power domino-resonator systems," *IEEE Transactions on Power Electronics*, vol. 27, no. 4, pp. 1905–1916, 2012.

[24] Y. Sun, F. X. Yang, and X. Dai, "Wireless power transmission network networking based on improved ant colony algorithm," *Journal of South China University of Technology*, vol. 39, no. 10, pp. 146–151, 2011, in Chinese.

[25] L. Xiang, Y. Sun, X. Dai et al., "Route optimization for wireless Power transfer network based on the ce method," in *Proceedings of the Electronics and Application Conference and Exposition*, pp. 630–634, IEEE, Piscataway, NJ, USA, 2014.

[26] Y. Sun, J. F. Xia, C. S. Tang, and L. J. Xiang, "Optimization of wireless power transmission network path using cellular genetic algorithm," *Journal of Xi'an Jiaotong University*, vol. 51, no. 4, pp. 30–36, 2017, in Chinese.

[27] R. C. Zhang and Y. H. Li, "C-band microwave wireless power transmission transmitter design," *Modern Electronic Technology*, vol. 40, no. 9, pp. 6–13, 2017, in Chinese.

[28] R. M. Dickinson, "Performance of a high-power, 2.388 GHz receiving array in wireless power transmission over 1.54 km," in *Proceedings of the 1976 IEEE-MTT-S International Microwave Symposium*, pp. 139–141, IEEE, Cherry Hill, NJ, USA, June 1976.

[29] Y. Song, Y. Liu, R. Wang, and M. Ming, "Multi-objective configuration optimization for isolated microgrid with shiftable loads and mobile energy storage," *IEEE Access*, vol. 7, pp. 95248–95263, 2019.

[30] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[31] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[33] K. Buayai, W. Ongsakul, and N. Mithulananthan, "Multi-objective micro-grid planning by NSGA-II in primary distribution system," *European Transactions on Electrical Power*, vol. 22, no. 2, pp. 170–187, 2012.

[34] X. Y. Lu, Y. Q. Huang, N. Liu, J. H. Zhang, and L. Z. Huang, "A study of optimal capacity configuration with multi-objective for islanded micro-grid," *Applied Mechanics and Materials*, vol. 521, pp. 464–468, 2014.

[35] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[36] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[37] R. Wang, R. C. Purshouse, and P. J. Fleming, "Whateverworks best for you" a new method for a priori and progressivemultibjective optimisation," in Lecture Notes in Computer Science, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds., Springer, Berlin, Germany, pp. 337 351, 2013.

[38] R. Wang, R. C. Purshouse, I. Giagkiozis, and P. J. Fleming, "The iPICEA-g: a new hybrid evolutionary multi-criteria decision making approach using the brushing technique," *European Journal of Operational Research*, vol. 243, no. 2, pp. 442–453, 2015.

[39] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1147–1154, IEEE, Beijing, China, 2014.

[40] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.