Research Article

A Multifactory Integrated Production and Distribution Scheduling Problem with Parallel Machines and Immediate Shipments Solved by Improved Whale Optimization Algorithm

Vahid Abdollahzadeh ¹, **Isa Nakhaikamalabadi** ², **Seyyed Mohammad Hajimolana** ¹,
and **Seyyed Hesamoddin Zegordi**³

¹Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

²Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran

³Department of Industrial Engineering, Tarbiat Modares University (TMU), Tehran, Iran

Correspondence should be addressed to Isa Nakhaikamalabadi; nakhaikamalabadi.i@wtiau.ac.ir

Received 26 April 2018; Revised 30 August 2018; Accepted 13 September 2018; Published 2 December 2018

Academic Editor: Roberto Natella

Copyright © 2018 Vahid Abdollahzadeh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper discusses the integrated scheduling of production and distribution operations in a multifactory supply chain with make-to-order production system. For the production side of the supply chain, we considered distributed parallel-established factories with identical parallel machines available at each factory. We assumed that the factories could produce all customers' orders with different production rates and costs. For the distribution side of the supply chain, we considered a limited number of homogeneous vehicles that immediately distribute the finalized orders to the customers. Then, a mixed-integer nonlinear programming model is developed to determine the detailed scheduling of production and distribution that minimizes the total costs of the supply chain including production, distribution, and late delivery costs. To solve the real-world scale problems, we developed a new whale optimization algorithm (WOA). Moreover, we conducted computational experiments by generating several test problems to evaluate the proposed algorithm. Statistical analysis showed that the proposed algorithm has better performance than traditional WOA for different scales of the problem. Moreover, it confirms the capability of the improved whale optimization algorithm (IWOA) to solve the medium-scale instances; however, the results indicate the better performance of genetic algorithm (GA) for the large-scale instances.

1. Introduction

The increasing competition in different marketplaces and the necessity of appropriate responses to the rapid changes in market requirements have made companies shift from a single production plant to multiple factories adjacent to their customers in different geographical areas. Such strategies reduce the transportation costs of the supply chain and allow companies to utilize economic privileges such as availability and low expense of raw materials, expert labors, energy, and tax allowances. The advent of new joint venture companies

in cooperation with global brands in developing countries is an indication of the fact.

Multifactory supply chains are classified in different categories such as ownership structure of factories, interaction mechanism between factories, and shop configuration. In some multifactory supply chains, all plants belong to the same company, while in some others, each factory belongs to different companies. For the first type, all factories work together to fulfill the objectives of the main company while in the second category, factories cooperate to maximize self-benefits. Different mechanisms of interactions in the factories throughout the

supply chain may lead to parallel, series, or network structures. Parallel factories can produce various types of products. In series structures, each production plant produces semifinished products for the next factory. The combination of parallel and series structures makes the network one [1]. Shop configuration in each factory can be considered the same as basic scheduling problem including single machine, parallel machine, flow shop, job shop, and open shop [2].

Numerous applications of the multifactory supply chains in the last two decades attracted the attention of researchers on multifactory scheduling problems in different industries such as electronics, food and chemical process industries, automotive companies, and electric power generating industries. Furthermore, integrating distribution and production scheduling operations as two costly functions of the supply chains made another subject of interest in applied operations research named integrated production and distribution scheduling problem (IPDSP). Independent optimization of the two following operations, as a traditional solution, would not necessarily lead to global optimality of the whole network. Thus, it is critical to consider interactions between production and distribution operations in the supply chain [3].

Besides, IPDSP in single factories was widely studied in the last decade. The researchers and practitioners extensively consider multifactory scheduling problem with different shop configurations and objective functions as maximizing total tardiness of the orders. Nevertheless, there is a limited amount of literature for the multifactory integrated production and distribution scheduling problem (MIPDSP) to minimize the total costs of the supply chain.

This paper is aimed at integrating the production and immediate distribution scheduling of various orders at parallel factories with identical parallel machines. The best allocation of the facilities to the orders along with the best scheduling of production and distribution pursue minimizing the operational costs including production, transportation, and late delivery. Therefore, the contributions of this paper are as follows:

- (i) Integrated mathematical formulation of production and immediate distribution scheduling in a parallel multifactory supply chain with a limited number of vehicles
- (ii) Considering trade-offs between late delivery, production, and transportation costs in make-to-order production systems to make decisions on the detailed scheduling of production and distribution
- (iii) Considering different orders with different amounts and due dates for any of the customers compatible with their dynamic lot sizing
- (iv) Considering the limited and equal fleet sizes at any of the factories with homogeneous vehicles
- (v) Improving newly proposed WOA to solve MIPDSP
- (vi) Introducing a new procedure used in the metaheuristic algorithms to ensure the feasibility of the solutions.

The remainder of this paper is divided into the following sections. Section (2) consists of a brief literature review. Section (3) defines the problem including assumptions and the mathematical formulation. Section (4) presents the solution approach consisting of a new proposed procedure used with metaheuristic algorithms and improved whale optimization algorithm (IWOA). Section (5) presents computational experiments' results. Conclusions and the trend of further researches are given in Section (6).

2. Literature Review

The existing papers regarding the scheduling of production and distribution operations can be grouped into three different categories: (i) IPDSP for single-factory supply chains, (ii) production scheduling problem in multifactory supply chains, and (iii) production and distribution scheduling problem in the multifactory environment. In this section, we review the recent papers proposed for each category in brief.

Chen [4] proposed the latest review article on IPDSP. They presented a unified model representation scheme and classified the existing models by the shipping and delivery methods. The IPDSP in the multifactory environment had been declared as a research gap by that time. In a more recent work on deterministic approaches of IPDSP, Guo et al. [5] presented a MINLP model for integrated production and distribution scheduling with parallel machines in a make-to-order supply chain. They developed a harmony search algorithm and a heuristic to solve the problem. Sawik [6] considered two inconsistent objectives including costs and service level in a three-echelon supply chain and addressed a bi-objective stochastic MIP model for joint selection of suppliers and scheduling of production and distribution for some numerical examples using the CPLEX solver. Cheng et al. [7] considered integrated scheduling of production and distribution problem to minimize the total cost of production and distribution. In the production side, there are capacitated batch-processing machines. In the distribution side, there is a third party logistic (3PL) provider to deliver the produce batches. They proposed a metaheuristic algorithm based on colony optimization method to solve the production scheduling and a heuristic algorithm for the distribution part to minimize total costs of scheduling for production and distribution operations in a single factory. Behnamian and Ghomi [2] discussed the significance of multifactory supply chains and for the first time provided a comprehensive review of the present papers.

A limited number of papers are available for the multifactory scheduling problem with parallel machines. Cicirello and Smith [8] presented a dynamic multifactory problem with parallel machines and applied wasp-like agents for distributed coordination of agents with parallel machines to decide on the acceptance of an arriving job on a machine queue. Hooker [9] combined MILP and constraint programming to solve the multifactory scheduling problem. They allocated the jobs to factories using MILP and scheduled on facilities using constraint programming (CP) and the two parts of the problem linked using logic-based Benders decomposition. Kerkhove and Vanhoucke [10] studied a real problem for the production

scheduling of knitted fabrics in a multifactory environment with unrelated single machines in each factory. They presented a hybrid metaheuristic, which combines simulated annealing and genetic algorithm to solve the large-scale problems. Behnamian and Ghomi [11] considered the multifactory problem with several parallel identical machines in each factory. The objective function for the problem was to minimize the maximum completion time of the jobs available at the beginning of the time horizon. They presented a MILP model for the problem as well as a heuristic and genetic algorithm for the solution. The presented that the modeling approach has been severely criticized by Yazdani et al. [12]. They analyzed and showed that the proposed model and related algorithm are suffering from serious shortcomings. They introduced three new models for minimizing makespan and total completion time in this paper. Furthermore, three efficient metaheuristics are put forward based on an artificial bee colony.

A limited literature is available for the integration of production and transportation scheduling with the serial multifactory structure. As the latest research, Karimi and Davoudpour [1] proposed a branch and bound algorithm for the scheduling of production and transportation problem in a serial multifactory production system. They also consider the same problem with batch delivery [13] so that the processed jobs should wait until all the jobs of the same batch are complete. Moreover, they developed their model and considered waiting time to start the process at the receiving factory [14]. A time-indexed formulation with an LP relaxation of binary variables is proposed; they considered deadlines for the same problem and suggested a new approach in the imperialistic competitive algorithm (ICA) to solve the large-scale problems. Terrazas-Moreno and Grossmann [15] considered a multiproduct integrated scheduling of production and distribution problem for the petrochemical industry. The partial production of demand is allowed in different complexes.

To the best of our knowledge, the single paper for the integrated scheduling of production and distribution problem in a multifactory environment with a make-to-order production system is proposed by Sun et al. [16]. They considered MIPDSP for a particular supply chain with two different transportation modalities: inland and maritime transportation. They also proposed a two-level genetic algorithm to solve the problem, one level for the assignment part of the problem and the other one for scheduling of the assigned jobs.

The MIPDSP studied in this research is strongly different from previous ones in supply chain structure and assumptions. The most appropriate production system for such supply chains is the make-to-order production system with immediate shipments, without any temporary warehouses and intermediate inventory. The real-world instances are project-based procurements in construction projects, automotive industry, knitting and clothing industry, dairy products, etc.

3. Problem Description

In this paper, we consider a supply chain with a set of parallel factories F , $F = \{1, \dots, f\}$ in different geographical locations

FIGURE 1: Supply chain structure.

owned by a holding company. The factories are controlled through a central planning system. There are m_α , $\alpha \in F$ parallel identical machines in each factory. Factories' production speeds are different due to technological differences. There are a set of geographically dispersed customers L , $L = \{1, \dots, l\}$. Each customer issues several orders with different sizes. The production cost and time of the orders are different at any of the factories. There are a limited number of capacitated vehicles to deliver the orders immediately after production. The order sizes are smaller than the vehicles' capacity. The orders' delivery cost is proportional to the type of products and traveling distance. An order can wait in the factory until its transportation starts without any holding cost. The vehicles deliver the orders to the customers on or later than their due dates, and the company incurs a time-dependent penalty for the late orders. Since all the factories belong to the same company, the most appropriate optimization objective is the minimization of operational costs including production, transportation, and late delivery costs. Figure 1 shows the supply chain configuration.

3.1. Assumptions

- (i) All parameters and required data are deterministic and available when scheduling is undertaken.
- (ii) There are orders from different customers available at time zero.
- (iii) The order sizes are smaller than the capacity of the vehicles.
- (iv) An equal number of homogeneous vehicles are available at each factory.
- (v) All factories can produce all the customers' orders.
- (vi) Each order can be processed at most on one machine, and any machine can process at most one order at any time.
- (vii) Factories can be idle, but machines cannot be idle when there are unprocessed orders in the plant.

TABLE 1: Definition of sets, indexes, parameters, and variables.

Sets	Definition
L	Set of customers
N	Set of orders
N_j	Order set for customer j
F	Set of factories
M_α	Machine set for factory α
Indexes	Definition
r, i	Order indexes
j, k	Customer indexes
α	Factory index
γ	Machine index
Parameters	Definition
$n_i/n_j/n_\alpha/n_\gamma$	Number of orders/batches/factories/machines
Q_{ij}	The size for order i of customer j
d_{ij}	The due date for order i of customer j
$\hat{\pi}_{ij}$	Late delivery cost for each unit of order i of customer j
p_{ija}	Processing time for order i of customer j at factory α
l_{ja}	Distance between customer j and factory α
vs	Mean vehicle speed in a unit of distance
c_{ij}^t	Transportation cost for order i of customer j per unit of distance
c_{ija}^p	Production cost for order i of customer j at factory α
Decision variables	Definition
$x_{ij,\alpha\gamma}$	1 if the order i of customer j is processed at factory α on machine γ ; 0 otherwise
$y_{ij,rk,\alpha\gamma}$	1 if the order i of customer j is processed before the order r of customer k at factory α on machine γ ; 0 otherwise
$s_{rk,\alpha\gamma}$	1 if the order r of customer k is scheduled as the first job at factory α on machine γ ; 0 otherwise
$e_{ij,\alpha\gamma}$	1 if the order i of customer j is processed as the last job at factory α on machine γ ; 0 otherwise
$s'_{rk,\alpha}$	1 if the order r of customer k is the first job delivered from factory α ; 0 otherwise
$e'_{rk,\alpha}$	1 if the order r of customer k is the last job delivered from factory α ; 0 otherwise
$z_{ij,rk,\alpha}$	1 if the order i of customer j is delivered before the order r of customer k at factory α ; 0 otherwise
ps_{ij}	Nonnegative variables for production start time of order i of customer j
ts_{ij}	Nonnegative variables for transportation start time of order i of customer j

TABLE 1: Continued.

Intermediate variables	Definition
P_{ij}	Nonnegative variables for production completion time of the order i of customer j
T_{ij}	Nonnegative variables for traveling time of the order i of customer j
C_{ij}	Nonnegative variables for delivering time of the order i of customer j
$MVP_{ij,rk}$	Maximum of production completion time and vehicle available time to transport order r of customer k when it is transported immediately after order i of customer j

- (viii) Different orders of a customer can be processed in various plants.
- (ix) Preemption is not allowed for production and delivery operations.
- (x) The vehicles cannot be idle when there are undelivered orders in a factory.

3.2. Mathematical Formulation. This section presents the mathematical formulation for MIPDSP which is the first step to understanding the characteristic of these problems. Table 1 presents the notations used in the model. A solution for MIPDSP includes (i) plants' allocation to the orders, (ii) the orders' production scheduling on machines, and (iii) transportation scheduling of orders from factories. We propose two lemmas used for simplification of the problem modeling.

Lemma 1. *The different numbers of available vehicles at various factories influence the assignment of jobs to the plants.*

Proof. Consider two factories f_1 and f_2 with v_1 and v_2 homogeneous vehicles and assume that $v_1 > v_2$. Consider only two orders (i, j) that belong to two different customers in different geographical positions with the same size and distance from any of the factories but with different due dates (d_i, d_j) and $d_i > d_j$. Consider that f_1 processes order j and f_2 processes order i . It means that the factory with more vehicles processes the most critical order. Changing the allocation of the factories will not improve the objective function so that the allocation of factories to the orders affects the objective function and rationally the factories with more available vehicles should produce the most critical orders.

Lemma 2. *The same numbers of available vehicles at each factory does not influence the assignment of the machines and sequencing of orders.*

Proof. Consider two factories f_1, f_2 and the orders (i, j) that belong to two different customers in different geographical positions with the same distance from any of the factories. Both of the orders have the same due date, and the

production and their delivery times are the same. The number of vehicles available at each factory is equal. Consider that f_1 processes order i and f_2 processes order j then changing the allocation of factories will not result in the worst objective function value. Therefore, the factories have the same priority in assigning the orders. Furthermore, if we allocate factory f_1 to process both of the orders and schedule order i before j , in this case, changing the sequence of the orders will not result in a worse objective function value. Hence, it is concluded that the equal number of vehicles has no influence on the sequencing and scheduling of orders to the machines.

Considering the lemmas, the optimal sequencing solution of the problem with the limited and equal number of homogeneous vehicles available at each factory without the loss of generality is the same as the optimal sequencing solution with single homogeneous vehicles available at each plant. For the simplification of the proposed model, we assume that there are single vehicles available at each factory and the vehicles at factories are homogeneous. The simplified MINLP model is as follows:

$$\begin{aligned} \text{Min} \quad & \sum_{j \in L} \sum_{i \in N_j} Q_{ij} \sum_{\alpha \in F} c_{ij\alpha}^p \sum_{\gamma \in M_\alpha} x_{ij,\alpha\gamma} \\ & + \sum_{j \in L} \sum_{i \in N_j} \sum_{\alpha \in F} l_{j\alpha} c_{ij}^t \sum_{\gamma \in M_\alpha} x_{ij,\alpha\gamma} \\ & + \sum_{j \in L} \sum_{i \in N_j} \hat{\pi}_{ij} Q_{ij} (C_{ij} - d_{ij}), \end{aligned} \quad (1)$$

$$\text{subject to} \quad \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} x_{ij,\alpha\gamma} = 1, \quad \forall j \in L, i \in N_j, \quad (2)$$

$$\sum_{j \in L} \sum_{i \in N_j} \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} (y_{ij,rk,\alpha\gamma} + s_{rk,\alpha\gamma}) = 1, \quad (3)$$

$$\forall k \in L, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} (y_{ij,rk,\alpha\gamma} + e_{ij,\alpha\gamma}) = 1, \quad (4)$$

$$\forall j \in L, i \in N_j, i \neq r \text{ or } j \neq k,$$

$$x_{ij,\alpha\gamma} x_{rk,\alpha\gamma} \geq y_{ij,rk,\alpha\gamma}, \quad \forall i \neq r \text{ or } j \neq k, \quad (5)$$

$$\sum_{j \in L} \sum_{i \in N_j} s_{ij,\alpha\gamma} \leq 1, \quad \forall \alpha \in F, \gamma \in M_\alpha, \quad (6)$$

$$\sum_{j \in L} \sum_{i \in N_j} e_{ij,\alpha\gamma} \leq 1, \quad \forall \alpha \in F, \gamma \in M_\alpha, \quad (7)$$

$$s_{ij,fm} \leq x_{ij,\alpha\gamma}, \quad \forall j \in L, i \in N_j, \alpha \in F, \gamma \in M_\alpha, \quad (8)$$

$$e_{ij,fm} \leq x_{ij,\alpha\gamma}, \quad \forall j \in L, i \in N_j, \alpha \in F, \gamma \in M_\alpha, \quad (9)$$

$$y_{ij,rk,\alpha\gamma} + y_{rk,ij,\alpha\gamma} \leq 1, \quad (10)$$

$$\forall j, k \in L, i \in N_j, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} (y_{ij,rk,\alpha\gamma} + y_{rk,ij,\alpha\gamma}) + s_{ij,\alpha\gamma} \leq 2, \quad (11)$$

$$\forall j \in L, i \in N_j, \alpha \in F, \gamma \in M_\alpha, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} (y_{ij,rk,\alpha\gamma} + y_{rk,ij,\alpha\gamma}) + e_{ij,\alpha\gamma} \leq 2, \quad (12)$$

$$\forall j \in L, i \in N_j, \alpha \in F, \gamma \in M_\alpha, i \neq r \text{ or } j \neq k,$$

$$P_{ij} = ps_{ij} + \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} Q_{ij} p_{ij\alpha} x_{ij,\alpha\gamma}, \quad \forall j \in L, i \in N_j, \quad (13)$$

$$ps_{ij} + M \left(1 - \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} s_{ij,\alpha\gamma} \right) \geq 0, \quad (14)$$

$$\forall j, k \in L, i \in N_j, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$M \left(1 - \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} s_{ij,\alpha\gamma} \right) - ps_{ij} \geq 0, \quad (15)$$

$$\forall j, k \in L, i \in N_j, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$ps_{rk} - ps_{ij} \geq \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} Q_{ij} p_{ij\alpha} y_{ij,rk,\alpha\gamma} - M \left(1 - \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} y_{ij,rk,\alpha\gamma} \right), \quad (16)$$

$$\forall j, k \in L, i \in N_j, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$ps_{rk} - ps_{ij} \leq \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} Q_{ij} p_{ij\alpha} y_{ij,rk,\alpha\gamma} + M \left(1 - \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} y_{ij,rk,\alpha\gamma} \right), \quad (17)$$

$$\forall j, k \in L, i \in N_j, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$\sum_{j \in L} \sum_{i \in N_j} \sum_{\alpha \in F} (z_{ij,rk,\alpha} + s'_{rk,\alpha}) = 1, \quad (18)$$

$$\forall \alpha \in F, k \in L, r \in N_k, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} \sum_{\alpha \in F} (z_{ij,rk,\alpha} + e'_{rk,\alpha}) = 1, \quad (19)$$

$$\forall \alpha \in F, j \in L, i \in N_j, i \neq r \text{ or } j \neq k,$$

$$z_{ij,rk,\alpha} + z_{rk,ij,\alpha} \leq 1, \quad (20)$$

$$\forall \alpha \in L, k \in L, r \in N_k, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} (z_{ij,rk,\alpha} + z_{rk,ij,\alpha}) + s'_{rk,\alpha} \leq 2, \quad (21)$$

$$\forall j \in L, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} (z_{ij,rk,\alpha} + z_{rk,ij,\alpha}) + e'_{rk,\alpha} \leq 2, \quad (22)$$

$$\forall j \in L, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$\sum_{j \in L} \sum_{i \in N_j} s'_{ij,\alpha} \leq 1, \quad \forall \alpha \in F, \quad (23)$$

$$\sum_{j \in L} \sum_{i \in N_j} e'_{ij,\alpha} \leq 1, \quad \forall \alpha \in F, \quad (24)$$

$$\sum_{k \in L} \sum_{r \in N_k} (z_{ij,rk,\alpha} + z_{rk,ij,\alpha}) \geq s'_{ij,\alpha}, \quad (25)$$

$$\forall j \in L, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$\sum_{k \in L} \sum_{r \in N_k} (z_{ij,rk,\alpha} + z_{rk,ij,\alpha}) \geq e'_{ij,\alpha}, \quad (26)$$

$$\forall j \in L, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$\left(\sum_{\gamma \in M_\alpha} x_{ij,\alpha\gamma} \right) \left(\sum_{\gamma \in M_\alpha} x_{rk,fm} \right) \geq z_{ij,rk,\alpha}, \quad (27)$$

$$\forall j \in L, k \in L, r \in N_k, i \in N_j, \alpha \in F, i \neq r \text{ or } j \neq k,$$

$$C_{ij} \geq ts_{ij} + T_{ij}, \quad \forall j \in L, i \in N_j, \quad (28)$$

$$T_{ij} = \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} \frac{l_{j\alpha}}{vs} x_{ij,\alpha\gamma}, \quad \forall j \in L, i \in N_j, \quad (29)$$

$$C_{ij} \geq d_{ij}, \quad \forall j \in L, i \in N_j, \quad (30)$$

$$ts_{ij} \geq P_{ij} - M \left(1 - \sum_{\alpha \in F} s'_{ij,\alpha} \right), \quad \forall j \in L, i \in N_j, \quad (31)$$

$$ts_{ij} \leq P_{ij} + M \left(1 - \sum_{\alpha \in F} s'_{ij,\alpha} \right), \quad \forall j \in L, i \in N_j, \quad (32)$$

$$MVP_{ij,rk} = \max \left\{ P_{rk} \sum_{\alpha \in F} z_{ij,rk,\alpha}, ts_{ij} + 2 \sum_{\alpha \in F} \sum_{\gamma \in M_\alpha} \frac{l_{j\alpha}}{vs} x_{ij,\alpha\gamma} \right\},$$

$$\forall j \in L, k \in L, r \in N_k, i \in N_j, i \neq r \neq k, \quad (33)$$

$$ts_{rk} \geq MVP_{ij,rk} - M \left(1 - \sum_{\alpha \in F} z_{ij,rk,\alpha} \right), \quad (34)$$

$$\forall j \in L, k \in L, r \in N_k, i \in N_j, i \neq r \text{ or } j \neq k,$$

$$ts_{rk} \leq MVP_{ij,rk} + M \left(1 - \sum_{\alpha \in F} z_{ij,rk,\alpha} \right), \quad (35)$$

$$\forall j \in L, k \in L, r \in N_k, i \in N_j, i \neq r \text{ or } j \neq k,$$

$$x_{ij,\alpha\gamma}, y_{ij,rk,\alpha\gamma}, z_{ij,rk,\alpha} \in \{0, 1\}, \quad (36)$$

$$s_{ij,\alpha\gamma}, e_{ij,\alpha\gamma}, s'_{ij,\alpha}, e'_{ij,\alpha} \in \{0, 1\}, \quad (37)$$

$$ps_{ij}, ts_{ij}, T_{ij}, P_{ij}, C_{ij}, MVP_{ij,rk} \in \mathbb{R}^+. \quad (38)$$

The objective function (i) minimizes the total production, transportation, and late delivery costs. The unit production costs for any of the orders are different in factories. For making the problem more realistic, it is assumed that transportation and late delivery costs of the product unit depend on the type of goods. The distribution cost of orders depends on the type of products and the transportation cost per unit of product and distance. The late delivery cost also depends on product type and delivery delay time. The first statement of the objective function represents the total production cost of the orders. The second statement defines total distribution cost, and the last statement shows the total late delivery cost.

Since each order should be processed in one factory and on one machine, Constraints (2) are established to assure it.

Constraints (3) ensure that any of the orders is a successor to another order or the first scheduled order on machines, and Constraints (4) ensure each order is the predecessor of another order or the last scheduled order on machines. Non-linear constraints (5) establish interconnections between machine allocation and orders scheduling subproblems. According to this constraint, if two different machines process two distinct orders, they should not be a successor or a predecessor to each other. Constraints (6) guarantee the selection of at most one order as first order on a machine, and Constraints (7) guarantee the same for the last order. Constraints (8) allocate a machine to the first scheduled jobs on the machines, and constraints (9) do the same for the last scheduled job on machines. Constraints (10) state that an order cannot be both predecessor and successor on a machine at the same time. Constraints (11) control the first scheduled order on a machine not to be preceded by the last scheduled order, and Constraints (12) control the last order of a production scheduling on a machine not to be preceded by the first scheduled order on the machine. Constraints (13) determine the production completion of orders, which is equal to the sum of the production start time of the order and its processing duration. Constraints (14) and (15) set the first order's production to start at the time zero. Constraints (16) and (17) ensure that the difference between production start times of two consecutive orders on the same machine is equal to the processing time of the successor. Constraints (18) assure that any order should be a successor or the first order delivered from a factory, and constraints (19) guarantee that any order should be a predecessor or the last order delivered from a factory. Constraints (20) ensure that any two orders transported from a plant not to be a predecessor and a successor to each other at the same time. Constraints (21) control the first scheduled order for transportation in a factory not to be preceded by the last scheduled order, and Constraints (22) control the last order of shipping schedule in a plant not to be preceded by the first order. Constraints (23) guarantee that there is at most one order at the beginning of transportation scheduling at each factory and, Constraints (24) guarantee at most one order at the end of transportation scheduling. Constraints (25) and (26) ensure that the orders selected as first and last order for transportation in a factory are in the transportation scheduling of the factory. Constraints (27) establish the interconnections between assignment and transportation scheduling subproblems. Constraints (28) calculate the delivery time of the orders that is the sum of transport start and travel times. Constraints (29) determine the traveling time between factory and customer proportional to the distance and vehicles' mean speed. Constraints (30) guarantee that the orders' delivery times are later than their due dates. Constraints (31) and (32) ensure the immediate transportation of produced orders. Constraints (33) determine feasible transportation start time. Transportation of an order should not start earlier than its production completion time and vehicle available time which is already transporting the previous order. Thus, these constraints determine the maximum of production completion time and the vehicle available time for the consecutively transported

FIGURE 2: Chromosome structure.

orders. Constraints (34) and (35) set the traveling start time of the orders. Constraints (36), (37), and (38) indicate the restrictions on binary and nonnegative variables including decision and intermediate variables.

4. Solution Approach

Since the assignment and parallel machine scheduling problems are both NP-hard problems, without loss of generality, the multifactory integrated scheduling of production and distribution problem is NP-hard as well [17, 18]. As a result, we propose a simulation-based intelligent procedure linked with metaheuristic algorithms to solve the large-scale problems. It is worth mentioning that every metaheuristic algorithm can be linked to the proposed procedure. In this research, recently proposed WOA is utilized to solve the scheduling problems for the first time. The helix-shape movement inspired from social behavior of humpback whale makes WOA different with existing algorithms. This mechanism provides a superior exploitation capability to the algorithm in solving the complex problems.

Furthermore, we improved the whale optimization algorithm which is called IWOA and evaluated its performance within the proposed procedure. We also evaluated the validity of the presented model and procedure by solving several random test problems in different sizes.

4.1. Solution Representation. The proposed problem includes several binary variables like the assignment of orders to factories and assignment of orders to machines. In this research, we use the random key method to represent the solutions. Each solution is represented by a linear chromosome including a certain number of genes with uniform random values in $[0,1]$. The length of the chromosomes is variable based on the number of customers, number of orders, number of factories, and number of machines. The structure of chromosomes is illustrated in Figure 2.

4.2. Simulation-Based Intelligent Procedure. Constraints satisfaction is a critical issue in metaheuristic algorithms for creating and updating the populations. MIPDSP is a complex problem that includes several interrelated subproblems and constraints that need to be feasible during the execution of metaheuristic algorithms. For example, a vehicle should ship an order from a factory that is produced in, an order's production should be scheduled in a plant that is assigned to, and the shipment of an order should be planned in the available time of vehicles. The penalty function is a traditional approach to deal with these kinds of constraints. Using penalty functions for the violating constraints does not necessarily lead to a final feasible solution. Thus, we proposed a four-step intelligent procedure based on random

simulation of operational situation and metaheuristic algorithms. In the proposed procedure, simulation of operations guarantees always feasible solutions and metaheuristic algorithms ensure intelligent search within the feasible solutions. Figure 3 illustrates the steps of the procedure.

Step 1. Initial random population

The proposed procedure is initiated with a random population in which the solutions are represented in random key alphabet. Each random solution in initial population is denoted by $u(i, j, \alpha, \gamma) \in [0, 1]$.

Step 2. Generate priority vectors

At this step, the procedure decodes the random variable $u(i, j, \alpha, \gamma)$ to a feasible solution $x(i, j, \alpha, \gamma) \in \{0, 1\}$ by determining the priority of orders to assign factories and machines. Algorithm 1 represents the pseudocode of decoding mechanism.

Step 3. Operation simulation

At this step, all the operational conditions including order assignment to factories, production scheduling on machines, and transportation scheduling are simulated according to priority vectors determined at Step 2. Intermediate variables like production completion times, delivery times, and delays are calculated accordingly.

Step 4. Fitness evaluation

The objective function is calculated at this step based on the main and intermediate variables. Moreover, the best solution within the population is determined if the stopping criterion was not satisfied.

Step 5. Evolutionary processes

At this step, evolutionary processes are implemented on the population to generate a new population to continue the procedure in Step 2.

To clarify the decoding mechanism, we proposed an illustrative example with two customers and two orders for each. We used Algorithm 1 to generate a random feasible solution to produce the orders at two factories with two parallel machines $(i, j, \alpha, \gamma = 1, 2)$. Consider the initial random solution which is represented in Figure 4. The algorithm reshapes the chromosome to $u(i, j, \alpha, \gamma)$ which is illustrated in Table 2. According to the algorithm, when $i = 1$ the sum should be 3 and when $i = 2$ the sum should be 3.6; on the other hand, when $j = 1$ the sum should be 2.7 and when $j = 2$ the sum should be 3.6. Thus, the algorithm adjusts the priority vectors $\vec{I} = [2, 1]$ and $\vec{J} = [2, 1]$. Then, the algorithm assigns random machines at random factories to the orders with the sequence (1) second order of customer two, (2) second order of the customer one, (3) then the first order of the customer two, and (4) first order of the customer one. We also used the same concept to determine the production and distribution sequencing of the orders.

4.3. Improved Whale Optimization Algorithm. The whale optimization algorithm is one of the newly nature-inspired metaheuristic algorithms, which is classified as swarm-

FIGURE 3: Intelligent procedure based on simulation and metaheuristic algorithms.

ALGORITHM 1: Decoding Mechanism (P^1).

0.3	0.2	0.1	0.6	0.3	0.1	0.4	0.7	0.6	0.1	0.7	0.4	0.3	0.5	0.2	0.8
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

FIGURE 4: An example chromosome.

agents update their position towards the best position using the following equations:

$$\vec{D}(t) = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|, \quad (39)$$

$$\vec{X}(t+1) = \vec{X}(t) + \vec{A}(t) \cdot \vec{D}(t), \quad (40)$$

based algorithms [19]. This algorithm mimics the social behavior of whales and their bubble-net hunting strategy. This algorithm assumes that the current best position is the target prey or is close to the optimal solution, so the other search agents will try to update their position towards the best search agent. This algorithm includes a new exploitation mechanism known as “bubble-net attacking method” and exploration mechanism called “search for prey.” The algorithm starts with a set of random solutions. At each iteration, search agents (whales) update their position towards either the best search agent or a randomly chosen agent. Search

where t indicates the iteration index, $\vec{A}$ and $\vec{D}$ are coefficient vectors, $\vec{X}^*$ is best position vector, and $\vec{A} \cdot \vec{D}$ is an element-by-element multiplication of two vectors. While updating the position of search agents, the algorithm makes a random selection between two approaches designed for the exploitation phase: (i) shrinking encircling mechanism and (ii) spiral

updating position. The following equations are designed to perform a random search by updating all search agents towards a random solution in the exploration phase.

$$\vec{D}'(t) = \left| \vec{C} \vec{X}_{\text{rand}}(t) - \vec{X}(t) \right|, \quad (41)$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}}(t) - \vec{A}(t) \cdot \vec{D}(t). \quad (42)$$

While conducting computational experiments using basic WOA, we observed some shortcomings in exploring solutions. Early convergence of WOA resulted in weak solutions compared with the other algorithms like GA and PSO. It seemed that the functions (41) and (42) designed for the random search of the algorithm are not able to ensure a good exploration of the solution region. To resolve the problem, we designed two perturbation mechanisms for both exploration and exploitation phases of the algorithm without disturbing the rationale behind the basic algorithm as follows:

- (a) On the exploitation phase, while using the shrinking mechanism, we considered adding a random vector to the search agents updated towards the best solution. We inspired this mechanism from the swarm-based firefly algorithm to increase exploration capability of WOA and diversification of solutions. We used (44) instead of (40) while updating the search agents using the shrinking mechanism in the improved algorithm.

$$\vec{D}(t) = \left| \vec{C} \vec{X}^*(t) - \vec{X}(t) \right|, \quad (43)$$

$$\vec{X}(t+1) = \vec{X}(t) - \vec{A}(t) \cdot \vec{D}(t) - \alpha \vec{e}(t). \quad (44)$$

- (b) On the exploration phase, besides using a random search agent for updating the current population, we considered a partial random movement considering the direction of the best search agent. The proposed improvement mechanism is designed to provide the algorithm with maximum exploration while updating the population. We used the following equations instead of (42) in our algorithm.

$$\begin{aligned} \vec{D}_d(t) &= \vec{X}^*(t) - \vec{X}(t), \\ \vec{X}(t+1) &= \vec{X}_{\text{rand}}(t) - \vec{A}(t) \cdot \vec{D}'(t) \\ &\quad + \vec{e}(t) \cdot \vec{D}_d(t), \end{aligned} \quad (45)$$

where $\vec{e}(t)$ in the third term of both equations is a vector of random numbers drawn from Gaussian or uniform distribution or any random vector according to the domain of the decision variables, and α is a constant number in $[0, 1]$ and $\vec{D}_d(t)$ is the vector from the current population towards the best search agent.

TABLE 2: Reshaped chromosome to $u_n(i, j, \alpha, \gamma)$.

i	j	α	γ	u
1	1	1	1	0.3
1	1	1	2	0.2
1	1	2	1	0.1
1	1	2	2	0.6
1	2	1	1	0.3
1	2	1	2	0.1
1	2	2	1	0.4
1	2	2	2	0.7
2	1	1	1	0.6
2	1	1	2	0.1
2	1	2	1	0.7
2	1	2	2	0.4
2	2	1	1	0.3
2	2	1	2	0.5
2	2	2	1	0.2
2	2	2	2	0.8

For detailed information on this algorithm and the equations for updating coefficients $\vec{A}$ and $\vec{D}$, you can refer to the recently published paper [19]. Algorithm 2 proposes the pseudocode of the IWOA using P^1 .

4.4. Solution Algorithm. The integrated nature of the problem with several interrelated subproblems creates a complex structure to MIPDSP, which results in too many constraint groups and decision variables. To confront this complexity, a new algorithm is proposed to solve the large-scale problems with respect to the proposed procedure. The steps of the algorithm are presented in Figure 5.

5. Computational Experiments

This section represents the results of computational experiments designed to examine the validity of the proposed solution approach using IWOA and other metaheuristic algorithms. We performed statistical analysis using one-way analysis of variance (ANOVA) technique to compare the performance of IWOA with WOA as well as different versions of two prevalent metaheuristic algorithms, GA and PSO.

5.1. Genetic Algorithm (GA). GA is an intelligent search algorithm that seeks optimal solutions for the NP-hard optimization problems. Furthermore, a standard and modified genetic algorithm has been used extensively to solve scheduling and routing problems [20–23]. Thus, standard GA with uniform crossover and mutation operators as well as a new version of GA with multiparent crossover operator (GA-MPC) [24] are utilized as measures to evaluate the efficiency of the proposed IWOA.

```

Input data
input  $l, b, \alpha$ 
Initialize
for  $n = 1$  to  $npop$ ;
generate  $u_n(i, j, f, m) \in [0, 1]$  for all  $i, j, f, m$ ;
decode the continuous vector  $\vec{u}_n$  to binary variable  $\vec{X}_n \in \{0, 1\}$  using  $(P^1)$ ;
evaluate the fitness of the initial population;
end;
save the best solution  $\rightarrow X^*$ ;
Main steps
while  $t \leq$  maximum number of iterations;
update  $(A, C)$ ;
update  $(p)$  as random variable  $\in [0, 1]$ ;
for  $n = 1$  to  $npop$ 
  if  $2p \leq 0.5$ 
    if  $2|A| \leq 1$ ;
    update the current search agent  $(\vec{u}_n(t))$  using improved shrinking encircling mechanism;
     $\vec{D}(t) = |\vec{C} \vec{u}^*(t) - \vec{u}(t)|$ 
     $\vec{u}(t+1) = \vec{u}(t) - \vec{A}(t) \cdot \vec{D}(t) + \alpha \vec{\epsilon}(t)$ 
  end if 2
  if  $2|A| > 1$ 
    select a random search agent from  $(\vec{u}_{rand}(t))$ ;
    update the current search agent using improved equations
     $\vec{D}'(t) = |\vec{C} \vec{u}_{rand}(t) - \vec{u}(t)|$ 
     $\vec{D}_d(t) = \vec{u}^*(t) - \vec{u}(t)$ 
     $\vec{u}(t+1) = \vec{u}_{rand}(t) - \vec{A}(t) \cdot \vec{D}'(t) + \vec{\epsilon}(t) \cdot \vec{D}_d(t)$ ;
  end if 2
end if 1
if  $3p > 0.5$ 
  update the current search agent  $(\vec{u}_n(t))$  using spiral updating mechanism;
end if 3
decode the continuous vector  $\vec{u}_n(t+1) \in [0, 1]$  to the binary variable  $\vec{X}_n(t+1) \in \{0, 1\}$  using  $(P^1)$ ;
implement the production and delivery scheduling considering the constraints;
evaluate the fitness of the  $\vec{X}_n(t+1)$ ;
end;
update the  $X^*$  if better;
 $t = t + 1$ ;
end while;
return  $X^*$ ;

```

ALGORITHM 2: Pseudocode for Improved WOA.

5.2. Particle Swarm Optimization (PSO). Since PSO is a swarm-based intelligent algorithm, it can be considered as a useful measure to evaluate the effectiveness of the proposed IWOA. Moreover, PSO is applied to solve several discrete optimization problems like machine scheduling, routing, and location-allocation problems [25–27]. Thus, canonical PSO and standard PSO 2011 called SPSO [28] are used within the proposed simulation-based algorithm (Figure 2) to evaluate IWOA for randomly generated instances.

5.3. Experiment Setup. Several experiments in different sizes including (i) small-scale, (ii) medium-scale, and (iii) large-scale instances are conducted to cover all the configurations of the problem. For our experiments, metaheuristic algorithms including GA, GA-MPC, PSO, SPSO, WOA, and

IWOA were coded in MATLAB (R2015b). Moreover, the linearized MINLP model was coded in the General Algebraic Modeling System (GAMS). Then, we generated several random instances for each size of the problem. Each experiment in this paper includes a random instance which is solved 30 times using each metaheuristic algorithm. Moreover, small and medium-scale instances are solved once with GAMS, CPLEX solver 12.2, restricting the run time to 500 seconds.

The experiments were performed on a Notebook PC Intel Core i7 (2.93 GH and 4 GB of RAM). We applied a design of experiments to tune the parameters of IWOA, PSO, GA, and GA-MPC which are represented in Table 3. Furthermore, we tuned the parameters of SPSO based on [28].

To evaluate the performance of the metaheuristic algorithms, we define three metrics: (i) percentage of deviation

FIGURE 5: Simulation-based algorithm.

TABLE 3: Parameters of metaheuristic algorithms.

Algorithm	Parameters	Small-scale	Medium- and large-scale
IWOA, WOA	N_{pop}	40	80
	Iterations	300	500
	b	6	10
GA (GA-MPC)	N_{pop}	40	80
	Iterations	200	300
	P_c	0.6 (0.5)	0.6 (0.5)
	P_m	0.4 (0.5)	0.4 (0.5)
PSO	N_{pop}	40	80
	Iterations	200	300
	W	0.3	0.3
	C_1	3	3
	C_2	3	3

(PD) as represented by (46), (ii) standard deviation (SD), and average of CPU times (ACT).

$$PD_a = \frac{Fit_a - Fit^*}{Fit^*} \times 100. \quad (46)$$

PD_a is the percentage of deviation of algorithm a , Fit_a is the fitness of algorithm a , and Fit^* is the best solution ever found for a random instance. For small-size instances, Fit^* is the optimal solution of the GAMS CPLEX solver, and for medium- and large-scale problems, it is the best solution ever found by metaheuristic algorithms or the GAMS solver. Since each experiment includes 30 observations, we used the mean value of PDs (MPD) to evaluate the metaheuristic algorithms' performances. Equation (47) represents MPD_a as

$$MPD_a = \frac{1}{30} \sum_{i=1}^{30} PD_{a_i}, \quad (47)$$

TABLE 4: Average of objective values for metaheuristic algorithms and mean PDs (%) for small-scale instances.

$i/j/f/m$	Number of variables	GAMS CPU time (s)	GAMS' objective value	IWOA	WOA	GA	GA-MPC	PSO	SPSO
				Average of obj. values	Average of obj. values	Average of obj. values	Average of obj. values	Average of obj. values	Average of obj. values
				MPD	MPD	MPD	MPD	MPD	MPD
2/2/2/2	112	0.353	4.61E+08	4.61E+08	0	4.61E+08	0	4.61E+08	0
2/2/2/2	112	0.326	1.36E+09	1.36E+09	0	1.36E+09	0	1.36E+09	0
2/2/2/2	112	0.312	7.24E+08	7.24E+08	0	7.24E+08	0	7.24E+08	0
2/2/2/2	112	0.308	5.72E+08	5.72E+08	0	5.72E+08	0	5.72E+08	0
2/2/2/2	112	0.423	2.43E+09	2.43E+09	0	2.43E+09	0	2.43E+09	0
2/2/2/2	112	0.321	3.16E+08	3.16E+08	0	3.16E+08	0	3.16E+08	0
2/2/2/2	112	0.354	1.95E+09	1.95E+09	0	1.95E+09	0	1.95E+09	0
2/2/2/2	112	0.236	8.63E+08	8.63E+08	0	8.63E+08	0	8.63E+08	0
2/2/2/2	112	0.325	7.49E+08	7.49E+08	0	7.49E+08	0	7.49E+08	0
2/2/2/2	112	0.247	1.38E+09	1.38E+09	0	1.38E+09	0	1.38E+09	0
3/2/2/2	240	67.4	1.77E+11	1.79E+11	1.6	1.79E+11	1.6	1.79E+11	1.6
3/2/2/2	240	196.3	1.79E+10	1.79E+10	0	1.79E+10	0	1.79E+10	0
3/2/2/2	240	20.59	2.83E+11	2.83E+11	0	2.83E+11	0	2.83E+11	0
3/2/2/2	240	171.03	8.82E+11	8.82E+11	0	8.82E+11	0	8.82E+11	0
3/2/2/2	240	393.95	1.32E+09	1.36E+09	3.5	1.36E+09	3.5	1.36E+09	3.5
3/2/2/2	240	58.43	4.80E+09	4.92E+09	2.3	4.92E+09	2.3	4.92E+09	2.3
3/2/2/2	240	110.52	7.31E+10	7.31E+10	0	7.31E+10	0	7.31E+10	0
3/2/2/2	240	93.44	6.41E+09	6.41E+09	0	6.41E+09	0	6.41E+09	0
3/2/2/2	240	213.67	9.32E+09	9.32E+09	0	9.32E+09	0	9.32E+09	0
3/2/2/2	240	43.72	1.42E+09	1.44E+09	1.23	1.44E+09	1.23	1.44E+09	1.23
2/3/2/2	240	68.56	4.81E+10	4.81E+10	0	4.81E+10	0	4.81E+10	0
2/3/2/2	240	234.79	4.59E+09	4.64E+09	1.04	4.64E+09	1.04	4.65E+09	1.21
2/3/2/2	240	62.99	5.30E+09	5.32E+09	0.4	5.32E+09	0.9	5.46E+09	2.1
2/3/2/2	240	26.5	1.52E+10	1.67E+10	8.9	1.69E+10	9.6	1.67E+10	8.9
2/3/2/2	240	229.23	3.38E+10	3.46E+10	2.4	3.46E+10	2.4	3.46E+10	2.4
2/3/2/2	240	279.3	2.37E+09	2.46E+09	3.4	2.46E+09	3.4	2.46E+09	3.4
2/3/2/2	240	95.34	1.69E+09	1.71E+09	1.03	1.73E+09	1.03	1.71E+09	1.03
2/3/2/2	240	162.49	3.47E+10	3.52E+10	1.45	3.52E+10	1.45	3.52E+10	1.45
2/3/2/2	240	348.26	2.04E+11	2.04E+11	0	2.04E+11	0	2.04E+11	0
2/3/2/2	240	279.36	1.60E+09	1.64E+09	2.12	1.64E+09	2.12	1.64E+09	2.12

TABLE 5: Mean PD(%), standard deviation ($SD \times 10^8$), average of CPU time (s), and gap of GAMS (%) for medium-scale instances.

$i/j/f/m$	Number of variables	IWOA			WOA			GA			GA-MPC			PSO			SPSO			GAMS	
		MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	PD	GAP
2/4/2/2	416	0.2	0.02	16	0.80	0.07	16	0.30	0.14	14	1.43	0.52	13	0.30	0.03	8	3.25	0.63	11	11.2	57.6
2/5/2/2	640	4.1	0.19	18	6.30	0.24	18	4.80	0.19	16	6.23	0.17	14	5.30	0.24	9	9.12	0.62	14	25.7	54.8
2/6/2/2	912	3.3	0.54	16	3.60	0.87	20	7.70	1.14	14	9.36	0.97	15	3.90	1.10	8	7.12	3.82	14	50.1	58.2
2/7/2/2	1232	3.8	0.83	23	5.01	1.40	22	4.40	0.84	19	7.84	0.73	18	5.40	1.24	11	9.70	4.85	18	60.4	61.3
2/8/2/2	1600	5.6	1.50	19	8.20	1.43	20	5.80	1.20	19	7.34	1.01	19	3.50	0.90	15	6.56	1.31	24	80.5	62.7
2/9/2/2	2016	4.4	1.20	21	8.70	4.10	22	7.20	1.70	22	11.5	2.45	19	4.20	2.15	13	8.22	5.32	23	80.7	63.5
2/10/2/2	2480	7.6	2.40	27	13.4	3.24	25	8.90	1.90	21	13.6	4.51	19	7.30	2.30	12	11.9	3.39	21	80.6	59.1
2/5/3/2	960	6.5	0.29	29	9.46	0.90	22	8.20	0.57	27	11.9	0.43	26	6.80	5.40	19	9.73	0.53	25	22.9	57.3
2/6/3/2	1368	6.2	0.67	22	12.1	0.89	23	9.20	0.89	20	13.2	1.73	20	5.70	0.99	14	9.34	1.82	23	39.1	58.3
2/7/3/2	1848	8.5	0.80	18	13.3	1.34	19	9.10	0.80	22	16.1	1.06	19	11.2	1.87	12	13.5	0.58	21	28.3	61.4
2/8/3/2	2400	6.8	2.20	26	11.7	3.8	26	10.9	2.40	25	16.3	4.23	23	10.3	2.20	22	15.5	5.36	27	83.5	62.7
2/9/3/2	3024	11.5	1.40	28	13.2	2.15	23	11.6	2.55	28	17.5	3.44	26	12.1	2.86	13	21.3	3.58	30	138	64.2
2/10/3/2	3720	12.8	1.80	31	21.5	3.16	25	11.5	1.66	25	14.1	3.49	18	9.50	2.26	14	15.3	7.84	26	158.3	63.6
3/3/2/2	522	1.97	0.31	18	2.99	0.35	19	2.36	0.39	20	7.34	0.67	19	2.45	0.28	10	6.13	0.56	17	10.8	59.4
3/4/2/2	912	5.10	0.86	20	5.75	0.95	19	5.81	1.01	19	10.6	1.14	18	4.69	1.02	11	7.84	1.23	13	54.3	56.7
3/5/2/2	1410	2.9	0.57	19	6.40	2.30	17	5.30	1.80	22	9.92	2.03	19	5.60	1.90	12	10.5	2.01	23	41.1	59.5
3/6/2/2	2016	4.3	0.85	20	6.13	1.90	20	3.90	1.42	19	11.1	4.48	17	3.20	1.45	10	7.88	4.35	19	50.5	60.2
3/7/2/2	2730	4.1	2.05	26	9.21	3.30	24	5.70	2.61	20	12.2	4.43	18	3.90	2.06	14	8.23	4.36	25	59.4	56.1
3/8/2/2	3552	5.7	3.70	25	11.1	4.36	24	7.50	2.90	23	12.7	4.84	20	6.30	4.00	12	12.9	5.71	28	81.3	57.4
3/9/2/2	4482	4.3	2.47	27	6.61	4.95	22	3.90	4.90	21	8.42	4.11	21	4.70	6.01	15	8.19	4.43	34	55.2	67.6
3/10/2/2	5520	10	6.72	45	19.5	11.2	57	10.7	9.33	39	14.1	9.60	30	10.5	8.13	16	18.4	13.9	30	78.5	62.3
3/10/3/2	8280	9.8	4.80	47	18.1	5.60	55	9.10	5.70	51	11.8	4.80	55	9.20	5.50	19	18.1	5.88	36	97.9	74.3
4/2/2/2	416	0.33	0.20	19	1.70	0.20	15	0.41	0.16	16	3.45	0.24	14	0.40	0.10	9	4.23	0.12	14	13.4	55.9
4/3/2/2	912	6.1	0.86	18	11.7	1.43	16	12.0	1.32	19	19.2	1.74	19	7.04	1.10	12	12.8	1.78	23	51.6	57.5
4/4/2/2	1600	4.4	0.86	27	9.70	1.86	22	7.40	1.43	24	15.5	2.31	21	4.10	1.10	11	11.9	1.68	20	59.4	55.4
4/5/2/2	2480	2.8	1.38	29	8.20	3.34	25	6.20	2.15	20	12.6	5.32	18	4.50	2.15	10	9.82	4.39	30	41.1	61.5
4/6/2/2	3522	4.1	1.25	27	7.50	2.32	26	6.40	3.55	25	4.71	12.2	20	3.40	1.33	14	5.32	2.27	19	67.7	62.8
4/7/2/2	4816	5.6	1.88	31	7.81	4.08	29	8.20	4.69	28	8.85	4.06	22	5.20	4.12	18	9.38	5.48	33	60.4	66.9
4/8/2/2	6272	8.04	4.11	47	12.7	8.51	41	6.10	4.20	32	11.2	6.94	27	6.70	7.70	9	13.4	10.2	24	80.2	68.4
4/9/2/2	7920	8.91	7.7	52	11.5	12.6	49	7.17	10.8	30	13.3	15.5	47	5.90	13.8	18	10.5	11.3	34	80.11	72.3
4/10/2/2	9760	9.40	10.6	58	11.9	16.3	45	11.8	14.0	49	13.4	15.8	47	8.20	20.7	17	10.2	11.4	31	106.8	89.1
4/5/3/2	3720	11.1	1.74	28	15.3	2.69	27	15.3	4.14	22	20.6	3.60	20	11.1	4.08	12	17.6	3.04	29	135.9	58.0
4/6/3/2	5328	11.3	3.58	44	12.7	4.09	39	11.7	4.09	32	10.9	3.88	28	8.56	3.01	15	19.2	3.02	40	180.8	69.3
5/2/2/2	640	1.60	0.23	15	5.05	0.33	16	4.93	0.34	16	7.7	0.65	17	4.97	0.48	9	5.47	7.08	17	14.89	57.3
5/4/2/2	2480	7.58	2.35	22	9.48	3.98	22	12.5	4.24	20	15.7	1.97	21	6.19	2.77	12	15.6	4.49	19	102.5	58.1

TABLE 5: Continued.

$i/j/f/m$	Number of variables	IWOA			WOA			GA			GA-MPC			PSO			SPSO			GAMS		
		MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	PD	PD	GAP
5/6/2/2	5520	5.79	3.6	39	10.3	5.1	33	8.59	8.56	29	11.4	4.79	23	8.41	9.29	17	13.1	6.91	38	78.8	66.0	
6/4/2/2	3552	8.46	3.00	26	15.1	4.64	24	7.71	2.09	22	15.2	2.88	19	5.97	3.03	13	14.3	3.37	32	98.7	61.2

FIGURE 6: MPD comparison of IWOA with WOA, GA, PSO, and GAMS for medium-scale instances.

TABLE 6: Results of one-way ANOVA test for medium-scale instances.

	DF	SS	MS	F-Value	P-Value	Test results
IWOA-WOA	1	270.3	270.34	16.21	0.00	Means are significantly different
Error	72	1200.9	16.68			
Total	73	1471.2				
IWOA-GA	1	40.89	40.89	3.51	0.065	Means are equal
Error	72	839.14	11.65			
Total	73	880.04				
IWOA-PSO	1	0.396	0.396	0.04	0.839	Means are equal
Error	72	686.03	9.528			
Total	73	686.43				
GA-PSO	1	33.25	33.25	3.20	0.078	Means are equal
Error	72	747.25	10.38			
Total	73	780.49				

DF: degree of freedom; SS: sum of squares; MS: mean squares.

where PD_{a_i} is the percentage of deviation for observation i of algorithm a .

In addition to graphical analysis of the results, one-way ANOVA test is conducted on the observations' MPDs for the statistical analysis of the results.

5.4. Results and Discussions. To evaluate the complexity of the problem, a single experiment is designed to solve a medium-scale random instance with 416 decision variables (2/4/2/2) using the GAMS CPLEX solver relaxing runtime

constraint. The best feasible solution obtained after a week runtime was just 9% better than the best solution found by metaheuristic algorithms in less than a minute. The results reveal the weakness of exact algorithms implemented by conventional software and the necessity of the computational methods to solve medium- and large-scale problems.

5.4.1. Comparison of Small-Scale Instances. For small-scale problems, 30 random numerical experiments are conducted considering all of the available configurations of the problem.

FIGURE 7: Boxplot diagrams for medium-scale instances.

FIGURE 8: SD comparison of IWOA with WOA and GA for medium-scale instances.

MPDs of these experiments are calculated using the optimal solutions obtained from the GAMS CPLEX solver in less than 500 seconds. The reasonable optimal solutions for the small-scale problems confirm the validity of the proposed MINLP model. Table 4 represents the comparative performance of different algorithms. Zeros in the MPD column of this table indicate the conformity of metaheuristic algorithms' solutions with the optimal solutions and insist the validity of metaheuristic algorithms as well. For small instances, while MPD of an instance is not zero, it means that there is at least one run among multiple runs of the algorithm that the solution of the metaheuristic algorithm was not equal to the optimal solution. Moreover, there was no experiment for IWOA with worse performance than other metaheuristic algorithms while there are two experiments for basic WOA that their MPDs are worse than other algorithms. Thus, we can conclude that the competence of IWOA

is at least the same as GA, GA-MPC, PSO, and SPSO for small-scale problems.

5.4.2. Results' Comparison for Medium-Scale Instances. To evaluate the performance of IWOA, 37 experiments are conducted for the medium-scale problems. The experiments' MPD, SD, and ACT are investigated in Table 5 for medium-scale instances. The last column of this table represents the deviation from best feasible solution and the gaps for solutions obtained from the GAMS CPLEX solver. Comparative analysis of GA and PSO with GA-MPC and SPSO indicated that MPDs of all the experiments for GA and PSO were better than their new versions. However, ACT of GA-MPC is better than GA and ACT of PSO is better than SPSO. Due to the significance of the solutions' quality in this research, traditional GA and PSO are used to evaluate IWOA for the small-scale problems.

TABLE 7: Mean PD(%), standard deviation ($SD \times 10^8$), and average of CPU time (s) for large-scale instances

$i/j/f/m$	Number of variables	IWOA			WOA			GA			GA-MPC			PSO			SPSO		
		MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT	MPD	SD	ACT
3/10/3/3	11070	10.4	3.10	59	20.3	3.52	50	8.04	3.73	48	11.2	3.63	40	17.1	4.50	20	10.7	3.52	38
3/10/3/4	13860	14.3	4.4	61	19.9	7.63	54	14.7	4.61	48	15.3	3.19	39	25.2	6.90	22	13.4	5.35	39
3/10/3/5	16650	17.7	3.79	65	24.3	3.77	60	17.2	1.75	51	10.1	1.89	44	27.2	4.42	23	14.0	3.73	45
4/7/3/2	7224	9.50	3.00	43	10.8	3.89	41	7.72	1.76	37	12.1	4.38	32	8.73	4.24	18	13.6	5.65	39
4/8/3/2	9408	11.4	5.07	40	15.2	4.60	37	8.62	2.85	35	10.2	5.63	34	9.71	9.31	15	11.8	3.95	34
4/10/2/3	13040	12.5	3.06	58	14.2	6.63	44	13.3	6.55	43	12.6	5.28	39	13.4	6.70	21	14.5	6.59	40
4/10/2/4	16320	13.1	3.79	63	17.3	5.64	58	12.4	13.4	46	11.6	9.00	35	21.1	7.20	25	14.1	9.76	43
4/10/2/5	19600	15.3	4.12	68	18.9	10.9	51	13.6	11.7	49	11.6	5.95	43	25.8	11.5	24	11.9	8.74	46
4/10/3/5	29400	24.2	4.49	80	26.2	7.03	74	17.6	5.22	63	16.4	5.17	54	38.2	7.61	30	22.9	5.79	64
4/10/4/5	39200	19.6	6.11	87	21.9	5.69	83	18.8	5.11	68	12.5	3.77	61	40.3	7.20	37	22.9	4.85	67
4/10/5/5	49000	26.1	6.39	107	29.1	5.32	99	14.6	4.71	79	10.6	4.18	63	42.5	5.32	42	32.8	5.17	101
5/7/3/2	11235	17.2	6.09	54	21.4	8.03	45	11.5	7.15	41	15.6	3.78	35	18.2	5.10	18	25.2	7.82	36
5/8/3/2	14640	12.1	9.70	57	17.3	7.39	49	11.6	8.62	43	12.1	6.27	38	15.3	9.05	21	12.1	8.17	37
5/10/4/2	30400	10.4	10.6	85	16.1	11.9	84	4.48	8.00	71	7.61	8.58	59	17.2	15.4	34	12.8	7.32	63
5/10/5/2	38000	16.7	4.54	91	18.6	11.2	83	15.8	12.6	76	10.2	6.00	68	20.9	17.4	36	18.8	9.62	69
5/10/5/3	50750	21.7	5.90	125	21.8	7.11	119	13.8	13.5	84	13.1	12.3	73	32.7	11.6	47	26.8	9.76	103
6/4/4/2	7014	15.8	2.90	39	21.1	2.90	44	11.7	1.38	38	28.3	1.19	30	12.3	2.42	17	27.1	2.84	40
6/8/2/2	14016	4.85	11.9	62	12.6	13.8	53	5.86	12.9	49	5.75	12.3	39	6.67	32.3	24	7.49	10.3	42
6/10/2/2	21840	9.01	13.2	71	10.5	21.1	64	6.64	16.2	50	7.11	27.1	44	6.68	19.9	26	10.5	23.2	55
6/10/5/2	54600	16.5	12.2	132	21.3	18.3	129	10.1	8.68	121	8.82	7.27	97	16.6	16.7	49	13.4	14.9	98
7/5/5/2	18725	12.6	5.94	67	13.2	4.83	52	15.2	37.8	45	11.4	5.60	42	19.5	8.80	23	15.3	6.88	48
7/8/4/2	38080	20.5	19.6	76	27.2	13.9	69	16.0	19.6	54	14.9	11.9	47	23.3	15.2	27	20.7	15.9	61
7/10/5/2	74200	21.1	18.7	153	26.2	25.1	149	8.81	23.6	131	8.12	21.5	104	26.1	17.7	57	20.8	21.2	107
8/9/3/2	47088	12.2	21.5	103	14.2	31.3	96	11.4	32.9	82	7.69	20.6	74	11.7	21.3	44	11.7	30.4	90
8/10/5/2	96800	19.6	16.9	192	20.4	19.0	183	11.4	25.6	139	10.3	14.4	112	21.3	18.8	66	21.2	13.7	122
10/5/2/3	20300	10.5	10.31	67	15.3	14.8	59	7.61	14.2	53	5.23	9.45	42	9.50	16.2	24	6.49	15.0	53
10/5/2/4	25400	12.2	9.67	72	12.7	6.29	71	11.1	7.37	66	5.84	10.6	55	14.4	8.85	28	11.3	11.3	69
10/5/2/5	30500	7.69	5.28	79	9.93	4.82	78	7.33	1.14	64	5.23	7.71	52	18.4	9.48	24	11.4	7.93	64
10/5/3/5	45750	17.1	4.63	99	19.1	5.83	91	9.70	8.3	82	7.81	6.99	74	36.1	9.39	34	20.7	6.92	61
10/5/4/5	61000	21.4	4.11	158	24.2	6.96	154	19.2	9.9	129	9.61	6.90	97	43.9	6.62	54	18.9	8.60	99
10/5/5/5	76250	23.1	6.50	167	24.1	7.78	159	12.5	5.6	136	10.1	5.28	109	43.3	9.43	56	20.6	7.84	104
10/10/2/2	60400	9.03	65.3	137	9.34	52.2	138	5.43	44.3	116	4.16	32.2	95	9.10	95.6	39	6.67	46.6	99
10/10/4/5	242000	17.8	12.9	382	22.5	19.5	413	8.33	20.4	341	4.48	18.4	316	35.2	22.7	90	31.3	19.4	189
10/10/5/5	302500	11.4	21.5	403	15.2	17.6	396	6.40	20.6	361	5.66	10.1	328	28.9	17.2	99	25.6	18.6	183
15/10/2/2	135600	7.67	120	241	9.83	112	285	6.17	102	237	5.32	96.2	216	8.16	125	75	6.84	120	153
20/10/2/2	240800	5.68	141	372	6.82	115	385	4.06	180	363	3.18	100	331	8.32	146	84	6.68	153	158
25/10/2/2	376000	4.59	272	397	6.25	353	410	2.73	250	382	3.02	207	370	5.10	229	98	4.92	190	176
30/10/2/2	541200	6.08	162	440	9.13	204	463	3.34	324	424	2.72	220	419	6.05	254	126	5.05	305	213
35/10/2/2	736400	4.95	433	315	7.96	630	341	2.28	377	294	2.27	237	275	5.11	477	98	4.21	442	195
40/10/2/2	961600	5.38	530	410	5.40	511	439	2.01	466	396	2.06	355	314	6.47	547	93	5.06	442	162
45/10/2/2	1216800	4.04	402	469	8.81	587	482	1.75	692	451	1.06	321	412	4.18	456	112	4.14	371	239
50/10/2/2	1502000	4.75	863	472	8.23	595	458	1.94	825	462	1.43	401	464	5.88	975	87	4.75	667	235

The graphical analysis of the results, as illustrated in Figure 6, implies that the performance of IWOA is better than WOA for all the instances considering the MPD matrix. On the other hand, the performance of IWOA is better than GA for 89.2% of the experiments, and also it is better than PSO for most of the instances.

Furthermore, one-way ANOVA test is conducted to compare IWOA with WOA as well as GA and PSO which had better performances. Although the graphical analysis illustrates a better performance for IWOA compared with GA and PSO, the results of ANOVA test on MPDs of these algorithms as illustrated in Table 6 and Figure 7 indicate

FIGURE 9: MPD comparison of IWOA with WOA, GA-MPC, SPSO, and GA-MPC with SPSO for large-scale instances.

TABLE 8: Results of the one-way ANOVA test for large-scale instances

	DF	SS	MS	F value	P value	Test results
IWOA-WOA	1	313.2	313.2	9.39	0.003	Means are significantly different
Error	82	2737.2	33.34			
Total	83	3047.4				
IWOA-GAMPC	1	400.2	400.16	12.62	0.001	Means are significantly different
Error	82	2599.3	31.70			
Total	83	2999.4				
IWOA-SPSO	1	47.82	47.82	0.98	0.325	Means are equal
Error	82	4006.18	48.86			
Total	83	4054.00				
GAMPC-SPSO	1	724.7	724.65	16.56	0.00	Means are significantly different
Error	82	3587.2	43.75			
Total	83	4311.8				

DF: degree of freedom; SS: sum of squares; MS: mean squares

similar competence for all three algorithms. Furthermore, the ANOVA test's results imply that MPD of IWOA is meaningfully better than WOA which confirms the graphical analysis results.

On the other hand, MPD of IWOA and nonzero PDs of the results obtained from the GAMS CPLEX solver which is illustrated in Figure 4 reveal the worse performance of

standard software and deterioration of solutions when the number of variables increases.

Figure 8 represents the comparison of SD for IWOA, WOA, and GA. Smaller SD of the experiments for IWOA confirms its improved robustness and stability of the solutions compared with WOA, GA, and PSO for the medium instances.

FIGURE 10: Boxplot diagrams for large-scale instances.

5.4.3. Results' Comparison for the Large-Scale Instances. Table 7 represents the results of the experiments on 42 large-scale instances to compare the performance of different metaheuristic algorithms. In our experiments, GA-MPC had better performance than GA considering both MPD and ACT metrics. Comparing MPDs of GA with GA-MPC indicated better performance of GA-MPC for 76.2% of the experiments. Furthermore, comparing PSO with SPSO with respect to MPD metrics revealed the better performance of SPSO for 88.1% of the test problems. However, ACT of PSO is better than SPSO. Due to the importance of solutions' quality, we have evaluated IWOA using GA-MPC and SPSO for the large-scale problems.

The graphical analysis of the results as illustrated in Figure 9, indicates that the performance of IWOA is better than WOA for all of the instances. Moreover, IWOA has better or same performance in 61.2% of the experiments compared with SPSO. However, the performance of GA-MPC is better than IWOA in 83.3% of the experiments. Furthermore, the comparative graphical analysis of IWOA and GA-MPC also reveals that the performance of IWOA has deteriorated as the number of decision variables increased.

On the other hand, one-way ANOVA test is conducted to compare the performance of IWOA with WOA as well as GA-MPC and SPSO which had better performances. The ANOVA test's results as investigated in Table 8 and Figure 10 indicate that the performance of IWOA is strongly better than that of WOA which confirms the results of the graphical analysis. Moreover, the ANOVA test's results show that IWOA has almost similar competence with SPSO while GA-MPC is more competent than other algorithms for the large-scale problems. However, IWOA reveals a

FIGURE 11: SD comparison of IWOA with WOA for large-scale instances.

FIGURE 12: SD comparison of IWOA with GA-MPC for large-scale instances.

similar performance compared with SPSO considering MPD of the observations.

Figures 11 and 12 represent almost the same performance for IWOA compared with WOA and GA-MPC with respect to solutions' SD.

6. Conclusion and Future Studies

In this study, we proposed a formulation for the integrated scheduling of production and distribution operations in a multifactory supply chain. Considering the make-to-order production system and direct shipment of finalized orders as well as the limited number of transportation vehicles increased the complexity of the problem. The integrated simulation-based solution approach that considers the interactions of subproblems also provides the best trade-off among the individual costs of the objective function. The advantage of our work is that it optimizes the scheduling of transportation integrated with production in the operational level of multifactory production systems. We used a conventional software to solve the small-size problems to evaluate the validity of the proposed model. We improved the newly introduced WOA to solve the medium-scale problems and computationally confirmed the efficiency of the modified algorithm (IWOA) compared to basic WOA for different scales of the problem. To evaluate IWOA, we compared the results of computational experiments with two versions of GA and PSO. The results of experiments confirmed a significant improvement in the mean value of the percentage of deviations while using IWOA rather than WOA for all sizes of the problem. However, unlike the improvements made to WOA and the better performance of IWOA for the medium-scale instances, it is better to use GA-MPC for the large-scale problems.

We considered parallel machines and immediate direct shipment of the orders in the problem of this research; other configurations of machines and shipping methods could be the trend of future studies. On the other hand, an integrated supply chain with a single objective function is considered in this paper. So, independent factories and multiobjective problem can be presented in the future. Furthermore, one can consider the application of the MIPDSP for the loading and distribution operations, and the general evaluation of IWOA as new directs of research.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] N. Karimi and H. Davoudpour, "A branch and bound method for solving multi-factory supply chain scheduling with batch delivery," *Expert Systems with Applications*, vol. 42, no. 1, pp. 238–245, 2015.
- [2] J. Behnamian and S. M. T. F. Ghomi, "A survey of multi-factory scheduling," *Journal of Intelligent Manufacturing*, vol. 27, no. 1, pp. 231–249, 2016.
- [3] Z. L. Chen and G. Pundoor, "Order assignment and scheduling in a supply chain," *Operations Research*, vol. 54, no. 3, pp. 555–572, 2006.
- [4] Z.-L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.
- [5] Z. Guo, L. Shi, L. Chen, and Y. Liang, "A harmony search-based memetic optimization model for integrated production and transportation scheduling in MTO manufacturing," *Omega*, vol. 66, Part B, pp. 327–343, 2017.
- [6] T. Sawik, "Integrated supply, production and distribution scheduling under disruption risks," *Omega*, vol. 62, pp. 131–144, 2016.
- [7] B. Y. Cheng, J. Y. T. Leung, and K. Li, "Integrated scheduling of production and distribution to minimize total cost using an improved ant colony optimization method," *Computers & Industrial Engineering*, vol. 83, pp. 217–225, 2015.
- [8] V. A. Cicirello and S. F. Smith, "Wasp-like agents for distributed factory coordination," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 237–266, 2004.
- [9] J. N. Hooker, "A hybrid method for the planning and scheduling," *Constraints*, vol. 10, no. 4, pp. 385–401, 2005.
- [10] L.-P. Kerkhove and M. Vanhoucke, "Scheduling of unrelated parallel machines with limited server availability on multiple production locations: a case study in knitted fabrics," *International Journal of Production Research*, vol. 52, no. 9, pp. 2630–2653, 2014.
- [11] J. Behnamian and S. M. T. Fatemi Ghomi, "The heterogeneous multi-factory production network scheduling with adaptive communication policy and parallel machine," *Information Sciences*, vol. 219, pp. 181–196, 2013.
- [12] M. Yazdani, S. Gohari, and B. Naderi, "Multi-factory parallel machine problems: improved mathematical models and artificial bee colony algorithm," *Computers & Industrial Engineering*, vol. 81, pp. 36–45, 2015.
- [13] N. Karimi and H. Davoudpour, "Integrated production and delivery scheduling for multi-factory supply chain with stage-dependent inventory holding cost," *Computational and Applied Mathematics*, vol. 36, no. 4, pp. 1529–1544, 2017.
- [14] N. Karimi and H. Davoudpour, "A knowledge-based approach for multi-factory production systems," *Computers & Operations Research*, vol. 77, pp. 72–85, 2017.
- [15] S. Terrazas-Moreno and I. E. Grossmann, "A multiscale decomposition method for the optimal planning and scheduling of multi-site continuous multiproduct plants," *Chemical Engineering Science*, vol. 66, no. 19, pp. 4307–4318, 2011.
- [16] X. T. Sun, S. H. Chung, and F. T. S. Chan, "Integrated scheduling of a multi-product multi-factory manufacturing system with maritime transport limits," *Transportation Research Part E: Logistics and Transportation Review*, vol. 79, pp. 110–127, 2015.
- [17] J. Bruno, E. G. Coffman, and R. Sethi, "Scheduling independent tasks to reduce mean finishing time," *Communications of the ACM*, vol. 17, no. 7, pp. 382–387, 1974.
- [18] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM*, vol. 23, no. 3, pp. 555–565, 1976.

- [19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [20] J. F. Gonçalves, J. J. de Magalhães Mendes, and Maurício G. C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem," *European Journal of Operational Research*, vol. 167, no. 1, pp. 77–95, 2005.
- [21] I. Essafi, Y. Mati, and S. Dauzère-Pérès, "A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 8, pp. 2599–2616, 2008.
- [22] H. E. Nouri, O. B. Driss, and K. Ghédira, "Hybrid metaheuristics for scheduling of machines and transport robots in job shop environment," *Applied Intelligence*, vol. 45, no. 3, pp. 808–828, 2016.
- [23] B. M. Ombuki and M. Ventresca, "Local search genetic algorithms for the job shop scheduling problem," *Applied Intelligence*, vol. 21, no. 1, pp. 99–109, 2004.
- [24] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 57–69, 2014.
- [25] M. Hasanzadeh, M. R. Meybodi, and M. M. Ebadzadeh, "Adaptive cooperative particle swarm optimizer," *Applied Intelligence*, vol. 39, no. 2, pp. 397–420, 2013.
- [26] B. Liu, L. Wang, and Y. H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 18–27, 2007.
- [27] K.-T. Fang and B. M. T. Lin, "Parallel-machine scheduling to minimize tardiness penalty and power cost," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 224–234, 2013.
- [28] M. Clerc, *Standard particle swarm optimisation from 2006 to 2011*, Tech. Rep. hal-00764996, 2011.

Submit your manuscripts at
www.hindawi.com