Research Article

An Optimization Framework of Multiobjective Artificial Bee Colony Algorithm Based on the MOEA Framework

Jiuyuan Huo ^{1,2} and Liqun Liu ³

¹School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

²Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou 730000, China

³College of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China

Correspondence should be addressed to Jiuyuan Huo; huojy@foxmail.com

Received 11 June 2018; Revised 10 September 2018; Accepted 27 September 2018; Published 1 November 2018

Academic Editor: Daniele Bibbo

Copyright © 2018 Jiuyuan Huo and Liqun Liu. 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.

The artificial bee colony (ABC) algorithm has become one of the popular optimization metaheuristics and has been proven to perform better than many state-of-the-art algorithms for dealing with complex multiobjective optimization problems. However, the multiobjective artificial bee colony (MOABC) algorithm has not been integrated into the common multiobjective optimization frameworks which provide the integrated environments for understanding, reusing, implementation, and comparison of multiobjective algorithms. Therefore, a unified, flexible, configurable, and user-friendly MOABC algorithm framework is presented which combines a multiobjective ABC algorithm named RMOABC and the multiobjective evolution algorithms (MOEA) framework in this paper. The multiobjective optimization framework aims at the development, experimentation, and study of metaheuristics for solving multiobjective optimization problems. The framework was tested on the Walking Fish Group test suite, and a many-objective water resource planning problem was utilized for verification and application. The experiment's results showed the framework can deal with practical multiobjective optimization problems more effectively and flexibly, can provide comprehensive and reliable parameters sets, and can complete reference, comparison, and analysis tasks among multiple optimization algorithms.

1. Introduction

The optimization problems in the real world are multiobjective in nature, which means that the optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. These problems are known as multiobjective optimization problems (MOPs) which can be found in many disciplines such as engineering, transportation, economics, medicine, and bioinformatics [1]. Most of the multiobjective techniques have been designed based on the theories of Pareto Sort [2] and nondominated solutions. Thus, the optimum solution for this kind of problem is not a single solution as in the mono-objective case, but rather a set of solutions known as the Pareto optimal set. This refers to when no element in the set is superior to the others for all the objectives.

By using the multiobjective optimization method, the conflicting objectives in these MOPs can acquire better trade-off, and satisfactory optimization results can be

given. Therefore, with the complexity and nonlinearity of objectives and constraints, finding a set of good quality nondominated solutions becomes more challenging, and research of efficient and stable multiobjective optimization algorithms is also one of the key and major directions for scholars to study. Over the last few decades, the metaheuristics algorithms [3] have proven to be effective methods for solving MOPs. Among them, the evolutionary algorithms are very popular and widely used to effectively solve complex real-world MOPs [4]. Some of the most well-known algorithms belong to this class, such as the Nondominated Sorted Genetic Algorithm-II (NSGA-II) [5], Multiobjective ϵ -evolutionary Algorithm based on ϵ Dominance (ϵ -MOEA) [6], and Borg [7].

Nevertheless, the swarm intelligence algorithm [8] inspired by biological information is one important type of metaheuristic algorithms. With its unique advantages and mechanisms, it has become a popular and important field. The main algorithms include the particle swarm optimization

(PSO) algorithm [9], ant colony optimization (ACO) algorithm [10], and shuffled frog leaping algorithm (SFLA) [11]. In 2005, Karaboga proposed an artificial bee colony (ABC) algorithm based on the foraging behavior of honeybees [12]. ABC has been demonstrated to have a strong ability to solve optimization problems, and its validity and practicality have been proven [13]. Because of achieving high convergence speed and strong robustness, it has been used in different areas of engineering and seems more suitable for multiobjective optimization. At present, the ABC algorithm and its application research mainly focuses on single-objective optimization. The study of multiobjective optimization has just begun.

However, because the multiobjective optimization needs to cope with real problems, there exists some inconvenience in practical applications. For instance, the multiobjective optimization algorithms are closely related to solving problems which are difficult to apply to other MOPs; a consistent model is needed to regulate and compare optimization strategies of different multiobjective optimization algorithms, and users have difficulty choosing the suitable optimization algorithm for their problems and also need to spend a lot of time learning the algorithms.

In this context, it is necessary to establish a unified, universal, and user-friendly multiobjective optimization framework which can be a valuable tool for understanding the behavior of existing techniques, for codes or modules that reuse existing algorithms, and for helping in the implementation and comparison of algorithms' new ideas. Moreover, researchers have found that focusing on the study of one algorithm has a lot of limitations. If different heuristic algorithms can be effectively referred or integrated with each other, they can handle actual problems or large-scale problems more effectively and more flexibly [14].

Therefore, multiobjective optimization frameworks have been proposed to integrate optimization algorithms, optimization problems, evaluation functions, improvement strategies, adjustment methods, and output of results to provide an integrated environment for users to easily handle optimization problems, such as the jMetal [15], Paradiseo-MOEO [16], and PISA [17]. Among them, the MOEA framework [18] is a powerful and efficient platform which is a free and open source Java library for developing and experimenting with multiobjective evolutionary algorithms (MOEAs) and other general purpose multiobjective optimization algorithms.

However, in these integrated environments for MO algorithms, the multiobjective artificial bee colony (MOABC) algorithm has not been integrated yet, and the MOABC algorithm has been proven in our previous research to perform better than many state-of-the-art MO algorithms [19]. Therefore, a multiobjective ABC algorithm named RMOABC [19] was introduced to integrate with the MOEA framework for providing a flexible and configurable MOABC algorithm framework that is independent of specific problems in this paper.

The remainder of this paper is organized as follows. The related literatures are reviewed in Section 2. Section 3 provides the background concepts and related technologies of MO and introduces the RMOABC algorithm. Section 4 described the unified optimization framework for MOABC algorithm based

on the MOEA framework. The case study is represented in Section 5. The experiment's settings, results, and corresponding analyses are discussed in Section 6, and finally, the conclusions and future work are drawn in Section 7.

2. Literature Review

In the past, methods based on metaheuristics developed by simulating various phenomena in the natural world have proven to be effective methods for solving MOPs [20]. Compared to traditional algorithms, modern heuristics are not tied to a specific problem domain and are not sensitive to the mathematical nature of the problem. They are more suitable for dealing with the practical MOPs. A subfamily of them in particular, the evolutionary algorithms, is now widely used to effectively handle MOPs in the real world [21]. In the mid-1980s, the genetic algorithm (GA) began to be applied to solve MOPs. In 1985, Schaffer [22] proposed a vector evaluation GA which realized the combination of the genetic algorithm and multiobjective optimization problems for the first time. In 1989, Goldberg proposed a new idea for solving MOPs by combining Pareto theory in economics with evolutionary algorithms, and it brought important guidance for the subsequent research on multiobjective optimization algorithms [23]. Subsequently, various multiobjective evolution algorithms (MOEAs) have been proposed, and some of them have been successfully applied in engineering [24]. For instance, Li et al. proposed a new multiobjective evolutionary method based on the differential evolution algorithm (MOEA/D-DE) to solve MOPs with complicated Pareto sets [25].

Since 2001, optimization algorithms based on swarm intelligence inspired by the cooperation mechanism of the biological populations have been developed [8]. Through the cooperation of intelligent individuals, the wisdom of the swarm can achieve breakthroughs beyond the optimal individual. Swarm intelligence algorithms have been successfully applied to handle the optimizing problems with more than one objective. For the multiobjective particle swarm optimization (MOPSO) [26] algorithms, a local search procedure and a flight mechanism that are both based on crowding distance are incorporated into the MOPSO algorithm [27]. Kamble et al. proposed a hybrid PSO-based method to handle the flexible job-shop scheduling problem [28]. Leong et al. integrated a dynamic population strategy within the multiple-swarm MOPSO [29].

Among the swarm intelligence algorithm, due to the high accuracy and satisfactory convergence speed, the ABC algorithm shows a greater advantage in problem representation, solving ability, and parameter adjustment [30]. Because research on the multiobjective ABC algorithm has just begun in recent years, there are relatively few studies on MOABC algorithms and its applications. For instance, Hedayatzadeh et al. designed a multiobjective artificial bee colony (MOABC) based on the Pareto theory and ϵ -domination notion [31]. The performance of Pareto-based MOABC algorithm has been investigated by Akbari et al., and the studies showed that the algorithm could provide competitive performance [32]. Zou et al. presented a multiobjective ABC that utilizes the Pareto-dominance concept

and maintains the nondominated solutions in an external archive [33]. And Akbari designed a multiobjective bee swarm optimization algorithm (MOBSO) that can adaptively maintain an external archive of nondominated solutions [34]. Zhang et al. presented a hybrid multiobjective ABC (HMABC) for burdening optimization of copper strip production that solved a two-objective problem of minimizing the total cost of materials and maximizing the amount of waste material thrown into the melting furnace [35]. Luo et al. proposed a multiobjective artificial bee colony optimization method called ε -MOABC based on performance indicators to solve multiobjective and many-objective problems [36]. Kishor presented a nondominated sorting based multiobjective artificial bee colony algorithm (NSABC) to solve multiobjective optimization problems [37]. Nseef et al. put forward an adaptive multipopulation artificial bee colony (ABC) algorithm for dynamic optimization problems (DOPs) [38]. In our previous works, a multiobjective artificial bee colony algorithm with regulation operators (RMOABC) which utilizes the mechanisms of adaptive grid and regulation operator was proposed in [19]. The experimental results show that compared with the traditional multiobjective algorithms, these variants of multiobjective ABC can find solutions with competitive convergence and diversity within a shorter period of time.

To effectively integrate different heuristic algorithms to handle MOPs more effectively and flexibly, a number of optimization algorithm frameworks were presented and applied in industrial and other fields. For instance, Choobineh et al. proposed a methodology for management of an industrial plant considering the multiple objective functions of asset management, emission control, and utilization of alternative energy resources [39]. Khalili-Damghani et al. proposed an integrated multiobjective framework for solving multiperiod portfolio project selection problems in the investment managers to make portfolio decisions by maximizing profits and minimizing risks over a multiperiod planning horizon [40]. An evolutionary multiobjective framework for business process optimization was presented by Vergidis et al. to construct feasible business process designs with optimum attribute values such as duration and cost [41]. Charitopoulos and Dua presented a unified framework for model-based multiobjective linear process and energy optimization under uncertainty [42]. Tsai and Chen proposed a simulation-based solution framework for tackling the multiobjective inventory optimization problem to minimize three objective functions [43]. A multiobjective, simulation-based optimization framework was developed by Avci and Selim for supply chain inventory optimization to determine supplier flexibility and safety stock levels [44]. Golding et al. introduced a general framework based on ACO for the identification of optimal strategies for mitigating the impact of regional shocks to the global food production network [45]. And a multiobjective optimization framework for automatic calibration of cellular automata land-use models with multiple dynamic land-use classes was presented by Newland et al. [46].

A number of multiobjective optimization framework for more general purposes have also been developed. For example, jMetal is an object-oriented Java-based framework designed to multiobjective optimization using metaheuristics

and is available to people interested in multiobjective optimization [47]. PISA is a C-based framework for multiobjective optimization which is based on separating the algorithm specific part of an optimizer from the application-specific part [17]. A framework for dynamic multiobjective big data optimization, jMetalSP combines the multiobjective optimization features of the jMetal framework with the streaming facilities of the Apache Spark cluster computing system that was presented to solve dynamic multiobjective big data optimization problems [48]. The MOEA framework [18] is a powerful and efficient platform that is a free and open source Java library for developing and experimenting with multiobjective evolutionary algorithms (MOEAs) and other general purpose multiobjective optimization algorithms.

In summary, the research of multiobjective ABC algorithms is still in the initial stage, and the MOABC algorithms are still not implemented in the common multiobjective optimization frameworks. Therefore, this paper focuses on introducing the RMOABC algorithm based on the Pareto dominance theory into the MOEA framework to establish a unified, universal, and user-friendly multiobjective optimization framework for the general optimization purpose.

3. Background Concepts and Related Technologies

3.1. Pareto Dominate Concepts. Multiobjective optimization often has to minimize/maximize two or more nonlinear objectives at the same time which are in conflict with each other. Thus, the trade-offs decisions should be taken between these objectives. Most of the multiobjective algorithms are proposed based on the Pareto Sort [2, 49] theory, so the optimization result is not usually a single solution but rather a set of solutions named as a Pareto nondominated set.

Generally, a multiobjective optimization problem is to optimize a set of objectives subjected to some equality/inequality constraints. The goal of multiobjective optimization is to guide the optimization process towards the true or approximate Pareto front and to generate a well-distributed Pareto optimal set. The basic concepts of the multiobjective method based on the Pareto theory can be found in [50].

3.2. Artificial Bee Colony Algorithm. The artificial bee colony (ABC) algorithm is a meta-heuristic and swarm intelligence algorithm proposed by Karaboga [12]. It is inspired by the foraging behavior of honeybees. Each individual bee is taken as an agent, and the swarm intelligence can be guided by the cooperation among different individuals. For its excellent performance, the ABC algorithm has become an effective means for solving complex nonlinear optimization problems.

The three types of bees—employed bees, onlookers, and scouts—constitute the artificial bee colony in the ABC algorithm. The optimization process is changed to the searching process of the nectar foods. Each position of the nectar source represents a feasible solution for the problem, and the nectar amount from the nectar source corresponds to the quality or fitness of the feasible solution. The evolutionary iterations and global convergence are achieved by

the cooperation of the three kinds of bees: (1) employed bees perform local random searches in the areas near their food sources; (2) onlookers make an optimum food source to further evolve in accordance with the specific mechanism; and (3) scouts update the stagnant food source according to the processing mechanism for stagnant solutions.

Overall, the employed bees and onlookers can work together to obtain better food sources through random and targeted searches. When the stagnant number of optimal food source reaches a certain value that is prone to fall into the local search, the scouts will start a new random exploration task for the global search. Thus, through the collaboration of the three kinds of bees, the ABC algorithm can quickly and effectively achieve global convergence.

3.3. RMOABC Algorithm. A typical goal in a multiobjective optimization problem is to obtain a set of Pareto optimal solutions. As identified earlier, it is necessary to provide a wide variety among the set of solutions for the decision-maker to choose from. By utilizing the Pareto theory, the original ABC algorithm has been improved and extended to handle the MOPs, and the new algorithm is called the RMOABC algorithm [19]. The RMOABC algorithm adopted two mechanisms, regulation operators and adaptive grid, to improve the accuracy and keep the diversity, respectively. And an external archive is also integrated to maintain the historical values of nondominated solutions found in the evolution process.

In the evolution process of optimization algorithms, it is essential to properly control the exploration and exploitation capabilities of the bees to efficiently find the global optimum for the optimization problem. According to the main update equation (i.e., Equation (1)) of the original ABC algorithm, it can be found that more emphasis is taken on the exploration capability [12]:

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}), \quad (1)$$

where x_{ij} (or v_{ij}) denotes the j -th element of X_i (or V_i); j is a random index; x_k denotes another solution selected randomly from the population; and ϕ_{ij} is a random number in $[-1, 1]$. It is well known that the exploration and exploitation capabilities of ABC heavily depend on the control parameters in the updated equation of the bees. Thus, to improve the exploitation capability of the ABC algorithm, Zhu et al. proposed a Gbest-guided artificial bee colony algorithm (GABC) in [51] to replace Equation (1) in the original ABC algorithm to obtain

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}) + \psi_{ij} * (y_j - x_{ij}), \quad (2)$$

where ψ_{ij} is a random number in $[0, 1.5]$ and y_j is the optimal solution fitness value in the j -th dimensional space.

To balance the trade-offs between the exploration and exploitation capabilities of MOABC, we proposed a multiobjective artificial bee colony algorithm with regulation operators (RMOABC) in [19] to dynamically adjust the capabilities of exploration and exploitation in the algorithm's evolution process. The local and global dynamic regulation operators were integrated with the GABC algorithm. The mechanisms are

to improve the ability of exploitation and guide the search of candidate solutions based on the information of global optimal solutions. The updated Equation (2) in the GABC algorithm was changed into the following equation:

$$v_{ij} = x_{ij} + k * \phi_{ij} * (x_{ij} - x_{kj}) + r * (y_j - x_{ij}), \quad (3)$$

where the local dynamic regulation operator k is set to $0.5 + \cos^2((\pi * i)/(2 * \text{MFE}))$, the global dynamic regulation operator r is set to $0.5 + \sin^2((\pi * i)/(2 * \text{MFE}))$, i is the current iteration number, and MFE is the maximum iteration number of algorithms. The details can be found in the literature [19].

In the design of multiobjective algorithms, the external archive is a typical method for maintaining the historical values of nondominated solutions found in the evolution process. The adaptive grid [52] mechanism proposed in the PAES (Pareto Archive Evolutionary Strategy) algorithm was utilized in the RMOABC to produce well-distributed nondominated Pareto solutions set in the external archive. Each nondominated solution can be mapped in a certain location in the grid according to the values of multiobjective functions. The grid can adaptively maintain the distribution of candidate solutions stored in the external archive in a uniform way in the evolution process. Thus, these two mechanisms adopted in the RMOABC algorithm can help the algorithm to quickly achieve global convergence. The details can be found in the literature [19].

3.4. MOEA Framework. Researchers of optimization algorithms have consensus that there is NO optimum strategy or algorithm for all of the optimized problems, but there is an effective strategy or algorithm for particular optimization problems. Thus, how to efficiently choose the proper algorithm for the particular optimization problem is a challenge for users. As mentioned above, a unified multiobjective optimization framework is good solution that can help users understand the behavior of existing techniques. And it can reuse the codes or modules in existing algorithms and can facilitate the implementation and comparison of new algorithms.

In multiobjective optimization frameworks, the MOEA framework is an open-source evolutionary computation library for Java that specializes in multiobjective optimization [18]. It is also an extensible framework for rapidly designing, developing, executing, and statistically testing multiobjective evolutionary algorithms (MOEAs). The framework supports a variety of state-of-the-art multiobjective evolutionary algorithms (MOEAs) such as NSGA-II (Nondominated Sorting Genetic Algorithm II), NSGA-III (Nondominated Sorting Genetic Algorithm III), ϵ -MOEA (Multiobjective ϵ -evolutionary Algorithm Based on ϵ Dominance), GDE3 (The Third Evolution Step of Generalized Differential Evolution), MOEA/D (Multiobjective Evolutionary Algorithm Based on Decomposition), PISA (Platform and Programming Language Independent Interface for Search Algorithms), and Borg MOEA. It also includes dozens of analytical test problems such as Zitzler-Deb-Thiele (ZDT), Deb-Thiele-Laumanns-Zitzler (DTLZ), CEC2009 (unconstrained problems), and so on. Thus, it can support the multiobjective optimization algorithm to be tested against a suite of state-of-the-art

Step 1: Initialization

Initialize parameters Θ^0 and solution set $X^0 \in S$, iteration index $k = 0$.

Step 2: Candidate generation

Generate candidate set $V^{k+1} \subset S$ according to generation rules and sampling distributions.

Step 3: Update solution/parameter set

Update solution set X^{k+1} according to candidate set V^{k+1} , previously obtained information and algorithm parameters. Update algorithm parameters Θ^{k+1} as well.

Step 4: Termination conditions

If termination conditions are satisfied, stop and exit; otherwise return to Step 2.

ALGORITHM 1: Unity procedure of metaheuristics.

algorithms across a large collection of test problems. The new problems can be conducted in numerous comparative studies to assess the efficiency, reliability, and controllability of state-of-the-art MOEAs.

4. The Unified Optimization Framework with MOABC Algorithm

The purpose of this paper is to present a unified optimization framework for the MOABC algorithm (UOF-MOABC), which combines the features of the MOEA framework [18] for multiobjective optimization metaheuristics with the RMOABC algorithm presented in [19]. Based on the advantages of a number of classic and modern state-of-the-art optimization, a wide set of benchmark problems and a set of well-known quality indicators assess the performance of the MO algorithms included in the MOEA framework. The UOF-MOABC can assist in multiobjective optimization research at the development, experimentation, comparison, and study of MOABC for solving multiobjective optimization problems.

4.1. System Architecture. The architecture of optimization algorithms should be generic enough to allow much needed flexibility to implement most of the metaheuristic; thus, before establishing the optimization framework, the metaheuristic should be characterized by a common behavior that is shared by all its algorithms.

As shown in Algorithm 1, unity procedures of metaheuristics were concluded by Wu et al. in [53]. This algorithm template is similar to most of the optimization algorithms that are based on the metaheuristic search which can be used to implement popular multiobjective technique and foster code reusability.

The MOEA framework has a number of algorithm templates that were summarized from the behavior of the base metaheuristic. Therefore, developing a particular algorithm only requires implementing the specific methods. And MOEA framework was designed based on the object-oriented architecture of Java that can facilitate the creation of new components and reusing of existing ones.

4.1.1. General Architecture of the MOEA Framework. The UML diagram, which includes the base classes within the MOEA framework, is summarized and depicted in Figure 1, which was acquired through in-depth analysis and research on

the source codes. To make the names of classes more general to be used in most of the metaheuristics, the MOEA framework adopted a generic terminology to name the classes. Therefore, the major components in the MOEA framework include *Algorithm*, *Problem*, and *Solution*. As shown in the figure, the working mechanism of MOEA framework is that an *Algorithm* is adopted to solve a *Problem* using a set of *Solutions*, the *Algorithm* performs the evolution process through a set of *Selection* and *Variation* operations, and the set of nondominated *Solutions* is assessed by the related evaluation methods. In the context of evolutionary algorithms, populations and individuals correspond to *Population* and *Solution* classes in the MOEA framework, respectively. The same classes can be applied to the ABC optimization algorithm related to the concepts of swarm and bees.

The class *Algorithm* represents the superclass for all the optimization metaheuristics. There are two main types of algorithms included in the MOEA framework. One is the native algorithms implemented within the framework that supports all functionality, and the other is the optimization algorithms provided by the JMetal library that can be executed within the MOEA framework. They are represented as the *AbstractAlgorithm* and the *JMetalAlgorithmAdapter* classes which are all inherited the class *Algorithm*, respectively. The classes can combine the optimization algorithm with the *Problem* (*getProblem()*) and the set of *Solutions* (*getResults()*) and evaluate the solutions with evaluation methods (*evaluation()*).

The class *Solution* represents a *Solution* object that is composed of an array of *Variable* objects. The class *Variable* is a superclass aimed at flexibly describing different kinds of representations for solutions that can contain variables of mixed variable types, such as *RealVariable*, *BinaryVariable*, *Program*, *Grammar*, and *Permutation*.

In the MOEA framework, all the problems have to inherit from interface *Problem* and class *AbstractProblem*. The interface *Problem* mainly contains two basic methods: *evaluate()* and *newSolution()*. The first method receives a *Solution* object representing a candidate solution to the problem and evaluates it. The second one is to generate a new *Solution* object for the problem according to the algorithm's mechanisms. The *Selection* and *Variation* interfaces aim at representing generic operators to be used by the different algorithms. The *Selection* operators include *TournamentSelection* and *UniformSelection*, and the *Variable* operators include *AdaptiveMultimethodVariation*, *CompoundCrossover*, *OnePointCrossover*, *TwoPointCrossover*, and *UniformCrossover*.

FIGURE 1: General architecture of the MOEA framework.

A more detailed description of the MOEA framework can be found in [18] and in the MOEA framework user manual.

4.1.2. Adding MOABC Algorithm to the Framework. This section is aimed at describing how the MOABC algorithm can be developed and included in the MOEA framework. And the RMOABC algorithm was taken as the demonstration case. To deal with this issue, the new algorithm must be adapted to comply with the standards of the MOEA framework. The UML class diagram of the MOABC algorithm and its variants which were implemented in the MOEA framework is depicted in Figure 2. To make the framework more versatile, we first designed a common class *MOABCAlgorithm* to implement the general attributes and methods of the MOABC algorithm. Through inheriting and implementing methods in the class *AbstractAlgorithm* of the MOEA framework, it achieves the integration of the MOABC algorithm with the

MOEA framework. Then, the implementation classes of the MOABC algorithm and its variant algorithms inherit the *MOABCAlgorithm* class and implement their own mechanisms by adding or modifying the attributes and methods.

Taking the RMOABC algorithm as a case, the class *RMOABCAlgorithm* needs to inherit the class *MOABCAlgorithm* and implements its specific methods such as *SendEmployedBees()*, *SendOnlookerBees()*, *SendScoutBees()*, and *UpdateExternalArchive()*. The Adaptive Grid mechanism of RMOABC produces well-distributed nondominated Pareto solutions set in the external archive, which is implemented by class *AdaptiveGrid* and class *Hypercube*.

After implementation of the MOABC algorithm in the MOEA framework, we began to describe how to execute it. The *Executor*, *Instrumenter*, and *Analyzer* classes provide most of the functionality provided by the MOEA framework [18]. The *Executor* class is responsible for constructing and executing runs of an algorithm. A single run

FIGURE 2: UML diagram of the MOABC algorithm and its variants.

requires three pieces of information: (1) the problem; (2) the algorithm used to solve the problem; and (3) the number of objective function evaluations allocated to solve the problem. The *Instrumenter* class works with the *Executor* class to record the necessary data while the algorithm is running. And the *Analyzer* class provides end-of-run analysis which is useful for statistically comparing the results produced by the different algorithms.

4.2. General Optimization Framework. Figure 3 gives an overview of the proposed UOF-MOABC framework for the general purpose optimization problems. As shown in the figure, the framework is comprised of four stages: *Problem Formulation*, *Algorithm Selection*, *Optimization Process*, and *Result Assessment* for RMOABC algorithm to solve the optimization problems.

4.2.1. Problem Formulation. In the *Problem Formulation* stage, the decision variables in this optimization problem and the specification of the objective function to be optimized, as well as any constraints on the decision variables or solutions need to be formulated according to the standards of the MOEA framework.

(1) Decision Variables. In the context of optimization, decision variables are unknown and controllable options that need to be determined in order to solve the problem; in other words, the problem is solved when the best values of the variables have been identified. The values of decision variables determine the values of the objective functions. The defining of decision variables is one of the hardest and

most crucial steps in formulating the optimized problem. Sometimes, a creative and suitable definition for the decision variables can dramatically reduce the size and difficulty of the problem. Thus, the MOEA framework provides different kinds of representations such as *RealVariable*, *BinaryVariable*, *Program*, *Grammar*, and *Permutation* for solutions which can flexibly contain variables of mixed variable types.

(2) Objective Functions. The objective function of an optimization problem indicates how much each decision variable contributes to the value to be optimized in the problem. As shown in Equation (4), the objective of the optimization process is to minimize or maximize the numerical value of the objective function by changing the values of selected decision variables:

$$\text{objective function} = f(x_1, x_2, \dots, x_i, \dots, x_n), \quad (4)$$

where $x = (x_1, x_2, \dots, x_i, \dots, x_n)$ is the vector of decision variables and x_i denotes the i -th decision variable in the n -dimensional decision space.

(3) Constraints. Constraints define the possible values the decision variables of an optimization problem may take. They typically represent resource constraints, or the minimum or maximum levels of an activity and take the general form as follows:

$$\begin{aligned} \text{subject to } & g_j(x) \leq 0, \quad j = 1, 2, \dots, J, \\ & h_k(x) = 0, \quad k = 1, 2, \dots, K, \end{aligned} \quad (5)$$

FIGURE 3: UOF-MOABC general optimization framework.

where $x = (x_1, x_2, \dots, x_i, \dots, x_n)$ is the vector of decision variables, x_i denotes the i -th decision variable in the n -dimensional decision space, $g_j(x) \leq 0$ defines J inequality constraints, and $h_k(x) = 0$ defines K equality constraints. Note that j is an index that runs from 1 to J , and each value of j corresponds to an inequality constraint. k is an index that runs from 1 to K , and each value of k corresponds to an equality constraint. The constraints generally reflect limitations of the real world system under consideration.

4.2.2. Algorithm Selection. Once the optimization problem has been formulated correctly, the optimization framework needs to choose the appropriate and suitable optimization algorithm for the problem in the *Algorithm Selection* stage. There are several configurations that need to be specified before running the optimization algorithm. Firstly, the parameters of selected optimization algorithms must be configured based on suggested values from the literature or

from previous experiences. These values of parameters influence optimization processes or search behaviors of the algorithm, such as population size, food source number, *Limit* number, and size of external archive in the case of the multiobjective ABC algorithm. Secondly, the termination criteria for the optimization process such as a predefined maximum iterations number or the precision value of no significant improvement in performance, must be specified. And finally, the number of execution times of the entire optimization process that has to be repeated must be specified to eliminate the randomness effects for the solutions, and to increase the chance for the optimal non-dominated solutions.

4.2.3. Optimization Process. The purpose of this stage is to use the selected optimization algorithm (such as the RMOABC algorithm) to identify the nondominated solutions for the optimization problem that provide the best possible trade-offs between the selected objective functions.

Then, the problem can be solved using the optimization algorithm in the following optimization processes. The solution generation mechanisms are used to generate trial solutions for the optimized problem from the search space of each decision variable. Various strategies can be adopted in the solution generation mechanisms to improve the computational efficiency and accuracy of the solutions. These trial solutions are then evaluated in terms of their constraints to verify whether they violate the constraints of the problem. And objective function values are also evaluated to judge whether the trial solution is better than the previous solution (for single objective problems) or whether it is a non-dominated solution to the problem (for multiobjective problem). Then, the algorithm judges whether the stopping criteria have been met, and, if so, the algorithm will output the resulting nondominated solutions; otherwise, information from the previous evaluation process is used to guide the generation of the next trial solutions using an optimization algorithm.

4.2.4. Result Assessment. Finally, in the last *Result Assessment* stage, the nondominated solutions for the optimized problem are quantitatively assessed and visualized. Then, the algorithm finishes the operation.

Many quantitative evaluation methods can be used to evaluate nondominated solutions to the optimal Pareto for the multiobjective optimization algorithms. These can mainly be divided into two categories: convergence and distribution [54]. The convergence indicator denotes the distance between the calculated noninferior front and the known true noninferior front or approximate true noninferior front. The distribution indicator refers to whether the obtained noninferior sets are evenly distributed in the optimal space. The most common used indicators are Generational Distance (GD) [55], Inverted Generational Distance (IGD) [56], Δ_p [57], Spacing (SP) [58], Hypervolume (HV) [59], and Computational Time (Times).

5. Test Problems

In this section, how the framework can be used for optimizing a practical problem will be discussed in detail. A well-known problem named the water resource planning (WRP) problem in the multiobjective literatures and the Walking Fish Group (WFG) test suite are taken as the test problems.

5.1. Water Resources Plan Problem. The water resource planning problem involves optimal planning for a storm drainage system in an urban area that was originally described by Musselman and Talavage in [60]. The problem entails examining a particular subbasin within a watershed. The subbasin is assumed to be hydrologically independent of other subbasins, having its own drainage network, on-site detention storage facility, treatment plant, and tributary to a receiving water body [61].

Mathematically, the WRP problem is a three-variable, five-objective, seven-constraint real-world problem to optimize the planning for a storm drainage system in an urban area. A detailed description of the problem can be found in [61], and it has been implemented in the MOEA framework.

5.1.1. Decision Variables. Three decision variables are assumed to characterize the storm drainage system of the subbasin: x_1 is the local detention storage capacity (unit: basin-inches), x_2 is the maximum treatment rate (unit: basin-inches/hour), and x_3 is the maximum allowable overflow rate (unit: basin-inches/hour) [61]. The model simulates the performance of the given storm drainage system which is defined by the variables, x_1 , x_2 , and x_3 over a specified period of time and under weather conditions representative of the area.

5.1.2. Objective Functions. There are five objective functions that should be minimized in the WRP problem. The f_1 function is the drainage network cost, f_2 function is the storage facility cost, f_3 function is the treatment facility cost, f_4 function is the expected flood damage cost, and f_5 function is the expected economic loss due to flood. The five objective functions are defined in the following equations:

$$\begin{aligned} f_1(x) &= 106780.37 * (x_2 + x_3) + 61704.67, \\ f_2(x) &= 3000 * x_1, \\ f_3(x) &= \frac{305700 * 2289 * x_2}{[0.06 * 289]^{0.65}}, \\ f_4(x) &= 205 * 2289 * e^{-39.75 * x_2 + 9.9x_3 + 2.74}, \\ f_5(x) &= 25 * \left(\frac{1.39}{x_1 * x_2} + 4940 * x_3 - 80 \right), \end{aligned} \quad (6)$$

where x_1 , x_2 , and x_3 are the three decision variables for the MRP problem.

5.1.3. Constraints. There are seven constraints that should be subjected in the WRP problem. The g_1 function is the average number of floods per year, g_2 function is the average flood volume per year, g_3 function is average number of pounds per year of suspended solids, g_4 function is the average number of pounds per year of settleable solids, g_5 function is the average number of pounds per year of biochemical oxygen demand, g_6 function is the average number of pounds per year of total nitrogen expected flood damage cost, and g_7 function is the average number of pounds per year of orthophosphate. The seven constraints are defined in the following equations:

$$\begin{aligned}
g_1(x) &= \frac{0.00139}{x_1 * x_2} + 4.94 * x_3 - 0.08 \leq 1.00, \\
g_2(x) &= \frac{0.000306}{x_1 * x_2} + 1.082 * x_3 - 0.0986 \leq 1.00, \\
g_3(x) &= \frac{12.307}{x_1 * x_2} + 49408.24 * x_3 + 4051.02 \leq 50000.00, \\
g_4(x) &= \frac{2.098}{x_1 * x_2} + 8046.33 * x_3 - 696.71 \leq 16000.00, \\
g_5(x) &= \frac{2.138}{x_1 * x_2} + 7883.39 * x_3 - 705.04 \leq 10000.00, \\
g_6(x) &= \frac{0.417}{x_1 * x_2} + 1721.26 * x_3 - 136.54 \leq 2000.00, \\
g_7(x) &= \frac{0.164}{x_1 * x_2} + 631.13 * x_3 - 54.48 \leq 550.00,
\end{aligned} \tag{7}$$

where x_1 , x_2 , and x_3 are the three decision variables for the MRP problem and $0.01 \leq x_1 \leq 0.45$, $0.01 \leq x_2 \leq 0.10$, and $0.01 \leq x_3 \leq 0.10$.

5.2. Walking Fish Group (WFG) Toolkit. The Walking Fish Group (WFG) toolkit [62] is a well-known continuous and combinatorial benchmark suite that can be scaled to any number of objectives and decision variables. Comprised of problems with various characteristics, such as linear, convex, concave, multimodal, disconnected, biased, and degenerated Pareto fronts, the WFG suite is used to challenge varying capabilities of MO algorithms. Their characteristics are summarized in Table 1 [63]. The parameters k and l in WFG are set to $(m - 1)$ and 10, respectively. m denotes the number of objectives, and then, the number of variables is set to $(m - 1) + 10$. In this paper, the objective number m is set to 2 and 3.

6. Experiments and Results Analysis

In this section, we describe the experimental study undertaken to evaluate the performance of the UOF-MOABC framework for solving the water resource planning (WRP) problem and the Walking Fish Group (WFG) toolkit.

6.1. Experimental Design. All of the experiments were performed on a PC with the Intel (R) Core i7-4720HQ @ 2.6 GHz 4 cores with 8 GB of RAM with Microsoft Windows 8 Professional Edition Version. We used the Java development kit (JDK) 1.7 and Eclipse 3.2 as the integrated development environment (IDE). And the version of the MOEA framework is 2.12.

6.1.1. Algorithm Selection. We intend to assess the performance of the RMOABC algorithm with six state-of-the-art

multiobjective algorithms which were implemented within the MOEA framework, such as the Nondominated Sorted Genetic Algorithm-II (NSGA-II) [5], Nondominated Sorted Genetic Algorithm-III (NSGA-III) [64], Multiobjective ϵ -evolutionary Algorithm Based on ϵ Dominance (ϵ -MOEA) [65], Speed-constrained Multiobjective PSO (SMPSO) [66], Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [67], and the third evolution step of Generalized Differential Evolution (GDE3) [68]. The experiments of simulation were taken on the UOF-MOABC framework which combines the MOEA framework with the RMOABC algorithm in the paper.

NSGA-II algorithm is the second generation of NSGA (Nondominated Sorted Genetic Algorithm) that addressed the deficiencies existing in the construction of nondominated set and the maintenance distribution strategy of a solution set. NSGA-III is the many-objective successor to NSGA-II, using reference points to direct solutions towards a diverse set. ϵ -MOEA utilizes the ϵ -dominance archiving for recording the Pareto optimal solutions; SMPSO is one of the multiobjective PSO algorithms and has the characteristic of limiting the velocity of the particles that will initiate generating the new effective particle positions if the velocity becomes too high. MOEA/D is an optimization algorithm based on the concept of decomposing the problem into many single-objective formulations. GDE3 is a multiobjective differential evolution (DE) algorithm for global optimization with an arbitrary number of objectives and constraints.

6.1.2. Assessment Methods. To allow a quantitative assessment and comparison of the performance of the selected multiobjective optimization algorithms, four indicators, Δ_p [57], SP [58], HV [59], and Times were adopted in the experiments. Δ_p is an averaged Hausdorff distance composed of GDp [69] and IGDp [69] and measures both diversity and spread [57]. And we take the most typical value $p = 2$ in this paper. The first three indicators are mainly used to evaluate the quality of the obtained Pareto solution set. The last indicator Times is the execution time of the optimization algorithm on the same computer which can reflect the time complexity of computation of the tested algorithm.

6.1.3. Parameter Settings. The selection of algorithm parameters can greatly affect the execution performance; thus, it is generally recommended to fine tune the parameters controlling the searching behavior of the optimization algorithm, such as population size, mutation probability, and crossover probability. Parameter calibration is a very time-consuming and computationally intensive work. Thus, most of the multiobjective algorithms adopted the recommended parameter settings from the MOEA framework in this paper.

The parameter settings of the seven multiobjective algorithms are shown in Table 2. D denotes the dimension of decision variables of the optimized problem. The population size is all set to 100, and the external archive capacity is set to 100. The stopping criterion adopted is reaching a certain number of generations, and the maximum evolution

TABLE 1: Properties of the Walking Fish Group (WFG) test problems.

Problem	Number of objectives (m)	Number of variables (n)	Properties
WFG1	2, 3	$(m - 1) + 10$	Mixed, flat biased
WFG2	2, 3	$(m - 1) + 10$	Convex, disconnected, nonseparable
WFG3	2, 3	$(m - 1) + 10$	Linear, degenerate, nonseparable
WFG4	2, 3	$(m - 1) + 10$	Concave, multimodal
WFG5	2, 3	$(m - 1) + 10$	Concave, deceptive
WFG6	2, 3	$(m - 1) + 10$	Concave, nonseparable
WFG7	2, 3	$(m - 1) + 10$	Concave, parameter dependent biased
WFG8	2, 3	$(m - 1) + 10$	Concave, nonseparable, parameter dependent biased
WFG9	2, 3	$(m - 1) + 10$	Concave, nonseparable, deceptive, parameter dependent biased

number of the algorithm is 10000 for WFG toolkit and is 2500 for WRP problem.

The multiobjective optimization algorithms mentioned above are all the nondeterministic techniques. Thus, it does not make much sense to draw some conclusion after running the algorithm once. The usual solution is to carry out a number of independent runs and then to use means and standard deviations for summarizing the obtained results. Thus, the results obtained from the 30 independent runs of each algorithm were statistically calculated to compare their performance.

6.2. Experimental Results for Walking Fish Group (WFG) Problems

6.2.1. Distribution Visualization of the Nondominated Solutions. To understand the distribution of the nondominated solutions obtained by the seven multiobjective algorithms in the visualization, Figure 4 plots the final nondominated solution set of the seven algorithms on the 3 objectives of WFG9, drawn by the parallel coordinates. Considering the length of the paper, and the WFG9 is the most complicated problem in the WFG test toolkit. Thus, we took it as the case for the study. The experiment result was obtained from the particular run for which its HV value of the result is the closest to the mean value. According to the definitions of the WFG test suite, the upper and lower bounds of the objective i of WFG test function are 0 and $2 * i$, respectively. Thus, the value ranges of objectives for 3 objectives WFG9 are [0, 2], [0, 4], and [0, 6].

Although all the considered nondominated solution sets from the seven algorithms appear to converge into the optimal front, the algorithms perform differently in terms of diversity maintenance, which can be seen from Figure 4. The final nondominated solutions set obtained by NSGA-II, NSGA-III, ϵ -MOEA, SMPSO, and MOEA/D are all not converged in the uniform distribution, and there are many apparent gaps in the range lines of the objective functions, which means the algorithms fail to reach some regions of the Pareto front. The solution sets of GDE3 and RMOABC seem to have a better uniformity and can almost cover all the regions of the Pareto front. And for the NSGA-III, ϵ -MOEA, and GDE3 algorithms, some of their solutions exceed the boundary of the WFG9 problem which can be found in the figures. It can be concluded from the above results that the

RMOABC algorithm can converge to the true Pareto front, have a better distribution of nondominated solutions, and appear to have a good covering of the whole Pareto front.

6.2.2. Optimization Process. To compare the convergence performance of the seven multiobjective algorithms for the WFG9 problem, the performance indicators of Δ_p , SP, and HV have been considered as the performance measures in this study. The variations of these three indicators over the iteration number of the seven algorithms are shown in Figure 5. The X axis is the iteration number of the algorithms, and the Y axis of Figures 5(a)–5(c) represents the values of Δ_p , SP, and HV of the solutions set of the algorithms, respectively.

It can be seen from the figure, for the Δ_p indicator, the seven algorithms perform well in the optimization process, especially the ϵ -MOEA, NSGA-III, and RMOABC algorithm. At about the 1200th iteration, the RMOABC algorithm exceeds other algorithms and outperforms others at the lower Δ_p value which means it has a better quality of the solutions. For the SP indicator, the MOEA/D algorithm does not perform well and has a fluctuating line and larger values. The NSGA-II, NSGA-III, SMPSO, and GDE3 perform similarly, and their SP values also fluctuate along the iterations. The ϵ -MOEA and RMOABC have the more stable evolution lines in the SP performance, and the former exceeds the latter at about 1100 iterations and they reach the similar value at about 9500 iterations. For the HV indicator, the NSGA-III, ϵ -MOEA, and RMOABC perform better than other algorithms, especially with the RMOABC algorithm exceeding the others at about 5000 iterations.

6.2.3. Multiobjective Performance Comparison. Through the thirty independent runs of the seven algorithms, the numerical statistical results in terms of the three performance metrics, Δ_p , SP, and HV and the computational time are shown in Tables 3–10. Max, Min, Mean, and SD represent the maximum value, minimum value, average value, and the standard deviation of the experiments results, respectively. The best mean among the algorithms for each WFG function is shown in bold and italics, respectively.

The statistical results of the Δ_p indicator for the WFG problems with 2 objectives are shown in Table 3. A lower value means better computed fronts; thus, we can see that the best or second best indicator values are distributed in three algorithms: NSGA-II, NSGA-III, SMPSO, and RMOABC. NSGA-

FIGURE 4: Plots of the final nondominated solution set of the seven algorithms on the 3 objectives of WFG9. (a) NSGA-II algorithm. (b) NSGA-III algorithm. (c) ϵ -MOEA algorithm. (d) SMPSO algorithm. (e) MOEA/D algorithm. (f) GDE3 algorithm. (g) RMOABC algorithm.

FIGURE 5: The performance indicators of Δ_p , SP, and HV vs. the iteration number of the seven multiobjective algorithms for the WFG9 problem with 3 objectives. (a) Δ_p indicator. (b) Spacing (SP) indicator. (c) Hypervolume (HV) indicator.

II, NSGA-III, and SMPSO have computed the best or the second best fronts regarding this indicator in two or three of the evaluated problems; RMOABC has obtained the best or second best values in this indicator for all the problems except the WFG1 and WFG9. For the SP indicator with the WFG problems of 2 objectives in Table 4, RMOABC can obtain the best or second best values in this indicator for all the problems but the WFG3 and WFG5; GDE3 got the best values for WFG5 and WFG7, and ϵ -MOEA got the best value for WFG3. For the HV indicator in Table 5, the larger the value means the better the quality of the solutions. RMOABC can obtain the best and second best values for all the problems but the WFG6; ϵ -MOEA got the best values for WFG5 and WFG6. And for the execution times indicator in Table 6, RMOABC can exceed other algorithms in most of the problems except the WFG1 and WFG2.

The Δ_p indicator's statistical results for the WFG problems with 3 objectives are shown in Table 7. It can be seen that the best or second best indicator values are distributed in four algorithms: NSGA-II, NSGA-III, ϵ -MOEA, and RMOABC. RMOABC has computed the best or the second best fronts regarding this indicator in most of the evaluated problems except WFG1 and WFG6. For the SP indicator with the WFG problems of 3 objectives in Table 8, ϵ -MOEA performs also very well in this indicator as it has obtained most of the best values for all the problems; RMOABC can obtain most of the second best values in this indicator for all the problems but the WFG2. For the HV indicator in Table 9, RMOABC can obtain the best and second best values for all the problems but the WFG4 and WFG5; NSGA-II got the best results for WFG3 and WFG4; ϵ -MOEA got the best result for WFG5. And for the execution

TABLE 2: The parameter settings of the seven multiobjective algorithms.

Algorithm	Parameter	Value	Remark
NSGA-II	sbx.rate	0.8	Crossover rate for simulated binary crossover
	sbx.distributionIndex	30.0	Distribution index for simulated binary crossover
	pm.rate	0.2	Mutation rate for polynomial mutation
	pm.distributionIndex	20.0	Distribution index for polynomial mutation
	withReplacement	True	Binary tournament selection
NSGA-III	Divisions	4	Number of divisions
	sbx.rate	0.8	Crossover rate for simulated binary crossover
	sbx.distributionIndex	30.0	Distribution index for simulated binary crossover
	pm.rate	0.2	Mutation rate for polynomial mutation
	pm.distributionIndex	20.0	Distribution index for polynomial mutation
ϵ -MOEA	sbx.rate	0.8	Crossover rate for simulated binary crossover
	sbx.distributionIndex	30.0	Distribution index for simulated binary crossover
	pm.rate	0.2	Mutation rate for polynomial mutation
	pm.distributionIndex	20.0	Distribution index for polynomial mutation
	Epsilon	0.1	ϵ values used by the ϵ -dominance archive
SMPSO	pm.rate	1.0/L	Polynomial mutation. L = individual length
	pm.distributionIndex	20.0	Distribution index for polynomial mutation
MOEA/D	de.crossoverRate	0.1	Crossover rate for differential evolution
	de.stepSize	0.5	Size of each step for differential evolution
	pm.rate	1.0/L	Mutation rate for polynomial mutation, L = individual length
	pm.distributionIndex	20.0	Distribution index for polynomial mutation
	neighborhoodSize	0.1	Neighborhood size used for mating
	Delta	0.9	Probability of mating
	eta	0.01	Maximum number of spots that an offspring can replace
GDE3	de.crossoverRate	0.1	Crossover rate for differential evolution
	de.stepSize	0.5	Size of each step for differential evolution
RMOABC	Adaptive grid number	25	
	<i>Limit</i>	$0.25 \times 100 \times D$	D denotes the dimension of decision variables

times indicator in Table 10, RMOABC can exceed other algorithms in most of the problems.

As shown in these tables, most of the algorithms can obtain a good solution set for solving the WFG problems, and the RMOABC and ϵ -MOEA algorithms perform the best for Δ_p , SP, and HV performance indicators, having a clear advantage over the other five algorithms on most of the test instances of two or three objectives. It demonstrated that the two algorithms are better than the other algorithms in terms of these performance indicators of convergence and distribution. Specifically, RMOABC can also obtain most of the best and second best results of execution times for WFG problems, and ϵ -MOEA needs to take several or even ten times longer to achieve similar results for the 3 objective problems. It demonstrated that the RMOABC can balance the various conflicting objectives and obtain a better performance than other algorithms.

6.3. Experimental Results for Water Resources Plan (WRP) Problem

6.3.1. Pareto Optimal Space (Range of Objective Values). The values range (i.e., the maximum and minimum) of the five objective functions for the water resources planning problem obtained by the seven algorithms is shown in Table 11. As can

be seen from the table, there are certain differences between the objective functions' results of the seven algorithms. This is the case because each algorithm has its own way to search the solution space, which may lead to different solutions for achieving similar assessment results.

For easy comparison and calculation, it is necessary to normalize the results of the objective functions. The specific normalized formula is shown in the following equation:

$$X_{\text{normalized}} = \frac{(X_{\text{original}} - X_{\min})}{(X_{\max} - X_{\min})}, \quad (8)$$

where X_{original} and $X_{\text{normalized}}$ are the original values of the objective function and the new value after normalization, respectively and $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the interval of the objective function. These were obtained from the Pareto front file in JMetal multiobjectives optimization framework [15], respectively. Ideally, the values ranges of the objective functions are normalized to $[0, 1]$.

The high-low-close chart is an efficient graphical tool for visual presentation of various forms of data area such as a range of measured values (min-max), 95% confidence interval value (low limit-high limit), and low value-average

TABLE 3: Statistical results of the Δ_p indicator obtained by different algorithms for the WFG problems with 2 objectives.

Functions	Δ_p	Algorithms						
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	6.067E - 01	7.170E - 01	6.473E - 01	6.579E - 01	4.920E - 01	6.225E - 01	7.057E - 01
	Min	4.795E - 01	5.466E - 01	6.215E - 01	6.133E - 01	4.612E - 01	5.835E - 01	6.362E - 01
	Mean	5.537E - 01	6.264E - 01	6.375E - 01	6.400E - 01	4.775E - 01	6.114E - 01	6.688E - 01
	SD	3.011E - 02	4.227E - 02	6.966E - 03	9.631E - 03	7.091E - 03	8.040E - 03	1.708E - 02
WFG2	Max	9.947E - 03	3.061E - 02	3.249E - 02	5.897E - 02	2.707E - 02	2.044E - 02	9.314E - 03
	Min	6.454E - 03	6.396E - 03	1.102E - 02	1.910E - 02	1.287E - 02	1.375E - 02	6.167E - 03
	Mean	7.506E - 03	1.026E - 02	2.193E - 02	3.474E - 02	1.759E - 02	1.682E - 02	7.283E - 03
	SD	8.957E - 04	5.599E - 03	5.465E - 03	1.079E - 02	3.149E - 03	1.771E - 03	6.871E - 04
WFG3	Max	3.764E - 02	3.772E - 02	4.460E - 02	4.971E - 02	3.925E - 02	8.769E - 01	3.734E - 02
	Min	3.469E - 02	3.483E - 02	3.451E - 02	3.941E - 02	3.574E - 02	7.769E - 01	3.412E - 02
	Mean	3.607E - 02	3.594E - 02	3.717E - 02	4.221E - 02	3.698E - 02	8.281E - 01	3.527E - 02
	SD	7.611E - 04	6.179E - 04	1.819E - 03	2.282E - 03	7.730E - 04	2.367E - 02	8.084E - 04
WFG4	Max	8.752E - 03	7.947E - 03	3.409E - 02	3.283E - 02	1.508E - 02	9.889E - 03	7.519E - 03
	Min	6.084E - 03	6.203E - 03	2.448E - 02	2.718E - 02	1.086E - 02	7.263E - 03	4.554E - 03
	Mean	7.622E - 03	6.960E - 03	2.887E - 02	3.073E - 02	1.270E - 02	8.734E - 03	5.955E - 03
	SD	6.306E - 04	4.394E - 04	1.916E - 03	1.408E - 03	1.039E - 03	5.439E - 04	8.199E - 04
WFG5	Max	2.952E - 02	2.768E - 02	2.805E - 02	2.784E - 02	2.999E - 02	2.723E - 02	2.739E - 02
	Min	2.794E - 02	2.732E - 02	2.732E - 02	2.721E - 02	2.764E - 02	2.716E - 02	2.725E - 02
	Mean	2.845E - 02	2.741E - 02	2.754E - 02	2.740E - 02	2.809E - 02	2.721E - 02	2.730E - 02
	SD	3.275E - 04	9.112E - 05	1.708E - 04	1.497E - 04	4.751E - 04	1.662E - 05	3.664E - 05
WFG6	Max	5.485E - 02	5.798E - 02	5.974E - 02	2.419E - 02	3.960E - 02	9.971E - 02	9.202E - 02
	Min	1.786E - 02	1.600E - 02	1.513E - 02	6.189E - 03	1.233E - 02	7.076E - 02	8.184E - 03
	Mean	3.583E - 02	3.345E - 02	3.882E - 02	2.023E - 02	2.614E - 02	8.605E - 02	2.194E - 02
	SD	9.524E - 03	8.892E - 03	1.230E - 02	5.886E - 03	6.652E - 03	6.550E - 03	2.035E - 02
WFG7	Max	7.862E - 03	7.393E - 03	1.028E - 02	2.294E - 02	9.589E - 03	5.893E - 03	3.903E - 03
	Min	5.856E - 03	5.524E - 03	5.191E - 03	1.100E - 02	7.396E - 03	4.827E - 03	2.797E - 03
	Mean	6.632E - 03	6.538E - 03	6.965E - 03	1.639E - 02	8.586E - 03	5.352E - 03	3.313E - 03
	SD	4.390E - 04	4.264E - 04	1.292E - 03	2.795E - 03	6.412E - 04	3.028E - 04	2.336E - 04
WFG8	Max	5.183E - 02	5.488E - 02	5.574E - 02	7.227E - 02	5.574E - 02	1.034E - 01	5.616E - 02
	Min	4.772E - 02	4.867E - 02	5.079E - 02	5.286E - 02	5.079E - 02	9.411E - 02	4.580E - 02
	Mean	4.979E - 02	5.153E - 02	5.301E - 02	6.299E - 02	5.301E - 02	9.782E - 02	4.946E - 02
	SD	1.079E - 03	1.723E - 03	1.282E - 03	5.396E - 03	1.282E - 03	2.444E - 03	2.577E - 03
WFG9	Max	2.148E - 02	1.965E - 02	1.017E - 01	1.604E - 02	1.032E - 01	3.266E - 02	1.578E - 02
	Min	1.006E - 02	9.546E - 03	7.329E - 03	9.397E - 03	1.001E - 02	9.777E - 03	9.608E - 03
	Mean	1.533E - 02	1.146E - 02	2.023E - 02	1.177E - 02	2.293E - 02	1.360E - 02	1.250E - 02
	SD	3.496E - 03	2.310E - 03	2.780E - 02	1.677E - 03	2.742E - 02	4.199E - 03	1.852E - 03

TABLE 4: Statistical results of the SP indicator obtained by different algorithms for the WFG problems with 2 objectives.

Functions	SP	Algorithms						
		NSGA-II	NSGA-III	ε -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	8.933E-01	5.589E-02	4.451E-02	5.944E-01	2.460E-01	8.224E-01	5.282E-02
	Min	6.785E-01	1.363E-02	1.167E-02	4.774E-01	1.988E-01	5.928E-01	1.141E-02
	Mean	7.813E-01	2.122E-02	2.309E-02	5.303E-01	2.304E-01	6.897E-01	2.048E-02
	SD	4.920E-02	8.176E-03	8.710E-03	3.291E-02	1.198E-02	5.364E-02	7.507E-03
WFG2	Max	7.803E-02	3.673E-02	3.502E-02	1.057E-01	1.388E-01	6.987E-02	1.788E-02
	Min	3.903E-02	1.929E-02	1.888E-02	3.141E-02	2.923E-02	2.648E-02	1.295E-02
	Mean	5.575E-02	2.377E-02	2.557E-02	5.235E-02	6.459E-02	4.812E-02	1.549E-02
	SD	9.663E-03	3.920E-03	4.182E-03	1.830E-02	2.165E-02	1.053E-02	1.376E-03
WFG3	Max	1.498E-02	2.200E-02	1.287E-02	1.803E-02	3.291E-02	1.553E-02	2.317E-02
	Min	8.530E-03	6.070E-03	9.160E-03	9.860E-03	2.222E-02	9.360E-03	1.535E-02
	Mean	1.089E-02	1.087E-02	1.059E-02	1.402E-02	2.599E-02	1.209E-02	1.593E-02
	SD	1.481E-03	3.454E-03	9.781E-04	1.970E-03	2.587E-03	1.477E-03	1.637E-03
WFG4	Max	9.755E-02	4.170E-02	4.944E-02	9.454E-02	6.011E-02	6.499E-02	3.345E-02
	Min	4.962E-02	2.205E-02	1.734E-02	2.027E-02	2.269E-02	3.509E-02	1.484E-02
	Mean	7.701E-02	2.750E-02	3.158E-02	5.391E-02	3.457E-02	4.684E-02	2.180E-02
	SD	1.044E-02	5.062E-03	7.654E-03	1.792E-02	1.029E-02	7.048E-03	4.040E-03
WFG5	Max	4.109E-02	2.376E-02	2.642E-02	1.835E-02	2.988E-02	1.579E-02	2.551E-02
	Min	2.224E-02	1.860E-02	1.682E-02	9.760E-03	2.394E-02	9.420E-03	1.862E-02
	Mean	3.109E-02	2.029E-02	2.093E-02	1.323E-02	2.603E-02	1.204E-02	2.181E-02
	SD	8.089E-03	1.568E-03	2.335E-03	2.001E-03	1.245E-03	1.472E-03	1.960E-03
WFG6	Max	6.226E-02	4.071E-02	4.147E-02	6.487E-02	3.234E-02	7.689E-02	2.759E-02
	Min	2.320E-02	2.259E-02	1.881E-02	1.151E-02	2.260E-02	2.366E-02	1.702E-02
	Mean	4.220E-02	2.753E-02	2.794E-02	2.259E-02	2.596E-02	4.937E-02	2.063E-02
	SD	8.941E-03	3.992E-03	6.439E-03	1.310E-02	2.325E-03	1.292E-02	2.721E-03
WFG7	Max	5.291E-02	4.059E-02	3.980E-02	4.875E-02	3.227E-02	3.576E-02	2.591E-02
	Min	2.754E-02	2.158E-02	2.095E-02	1.416E-02	2.328E-02	1.202E-02	1.857E-02
	Mean	3.995E-02	2.645E-02	2.987E-02	2.831E-02	2.672E-02	2.034E-02	2.274E-02
	SD	5.877E-03	3.694E-03	4.590E-03	9.755E-03	1.970E-03	5.534E-03	2.023E-03
WFG8	Max	5.703E-02	9.466E-02	5.342E-02	6.918E-02	6.547E-02	9.265E-02	3.139E-02
	Min	3.336E-02	2.278E-02	1.896E-02	1.474E-02	2.526E-02	2.893E-02	1.658E-02
	Mean	4.493E-02	3.385E-02	2.880E-02	3.085E-02	3.604E-02	5.772E-02	2.145E-02
	SD	5.287E-03	1.772E-02	8.599E-03	1.032E-02	9.450E-03	1.479E-02	3.257E-03
WFG9	Max	4.272E-02	2.919E-02	3.841E-02	2.287E-02	5.515E-02	2.059E-02	2.072E-02
	Min	1.881E-02	2.045E-02	1.501E-02	1.144E-02	2.463E-02	1.239E-02	1.361E-02
	Mean	2.899E-02	2.347E-02	2.445E-02	1.586E-02	2.831E-02	1.985E-02	1.522E-02
	SD	7.041E-03	2.106E-03	5.874E-03	2.602E-03	5.595E-03	2.148E-03	2.303E-03

TABLE 5: Statistical results of the HV indicator obtained by different algorithms for the WFG problems with 2 objectives.

Functions	HV	Algorithms						RMOABC
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3	
WFG1	Max	1.445E - 01	1.049E - 01	0.000E + 00	0.000E + 00	1.391E - 01	0.000E + 00	1.226E - 01
	Min	1.034E - 01	5.463E - 02	0.000E + 00	0.000E + 00	1.216E - 01	0.000E + 00	3.658E - 02
	Mean	8.867E - 02	8.099E - 02	0.000E + 00	0.000E + 00	1.306E - 01	0.000E + 00	1.202E - 01
	SD	1.234E - 02	1.457E - 02	0.000E + 00	0.000E + 00	4.708E - 03	0.000E + 00	1.876E - 02
WFG2	Max	5.606E - 01	5.599E - 01	5.491E - 01	5.410E - 01	5.545E - 01	5.520E - 01	5.615E - 01
	Min	5.537E - 01	5.452E - 01	5.260E - 01	4.988E - 01	5.455E - 01	5.378E - 01	5.533E - 01
	Mean	5.579E - 01	5.552E - 01	5.385E - 01	5.202E - 01	5.503E - 01	5.438E - 01	5.586E - 01
	SD	1.648E - 03	3.042E - 03	6.220E - 03	1.061E - 02	2.209E - 03	3.172E - 03	1.723E - 03
WFG3	Max	4.388E - 01	4.387E - 01	4.385E - 01	4.332E - 01	4.378E - 01	4.385E - 01	4.398E - 01
	Min	4.336E - 01	4.317E - 01	4.320E - 01	4.154E - 01	4.296E - 01	4.344E - 01	4.358E - 01
	Mean	4.370E - 01	4.365E - 01	4.352E - 01	4.263E - 01	4.350E - 01	4.359E - 01	4.379E - 01
	SD	1.358E - 03	1.413E - 03	1.954E - 03	3.980E - 03	1.504E - 03	9.408E - 04	1.047E - 03
WFG4	Max	2.142E - 01	2.153E - 01	1.853E - 01	1.848E - 01	2.082E - 01	2.125E - 01	2.175E - 01
	Min	2.105E - 01	2.124E - 01	1.777E - 01	1.778E - 01	1.977E - 01	2.076E - 01	2.113E - 01
	Mean	2.124E - 01	2.138E - 01	1.821E - 01	1.808E - 01	2.036E - 01	2.097E - 01	2.142E - 01
	SD	9.463E - 04	8.231E - 04	2.052E - 03	1.503E - 03	2.283E - 03	1.250E - 03	1.555E - 03
WFG5	Max	1.950E - 01	1.955E - 01	1.976E - 01	1.962E - 01	1.945E - 01	1.963E - 01	1.975E - 01
	Min	1.927E - 01	1.950E - 01	1.971E - 01	1.954E - 01	1.858E - 01	1.961E - 01	1.965E - 01
	Mean	1.940E - 01	1.953E - 01	1.974E - 01	1.959E - 01	1.938E - 01	1.962E - 01	1.972E - 01
	SD	5.273E - 04	1.348E - 04	1.067E - 04	2.195E - 04	1.554E - 03	4.417E - 05	2.774E - 04
WFG6	Max	1.827E - 01	1.917E - 01	2.041E - 01	2.040E - 01	1.883E - 01	1.881E - 01	1.882E - 01
	Min	1.518E - 01	1.447E - 01	1.153E - 01	1.801E - 01	1.539E - 01	1.608E - 01	1.475E - 01
	Mean	1.663E - 01	1.677E - 01	1.862E - 01	1.857E - 01	1.723E - 01	1.733E - 01	1.679E - 01
	SD	9.183E - 03	1.006E - 02	2.412E - 02	6.879E - 03	9.347E - 03	6.877E - 03	1.101E - 02
WFG7	Max	2.065E - 01	2.069E - 01	2.055E - 01	1.982E - 01	2.036E - 01	2.075E - 01	2.100E - 01
	Min	2.046E - 01	2.039E - 01	1.989E - 01	1.843E - 01	2.007E - 01	2.057E - 01	2.086E - 01
	Mean	2.055E - 01	2.056E - 01	2.031E - 01	1.915E - 01	2.022E - 01	2.067E - 01	2.093E - 01
	SD	5.367E - 04	7.907E - 04	1.849E - 03	3.531E - 03	7.302E - 04	4.868E - 04	3.846E - 04
WFG8	Max	1.604E - 01	1.599E - 01	1.630E - 01	1.539E - 01	1.589E - 01	1.574E - 01	1.631E - 01
	Min	1.533E - 01	1.564E - 01	1.479E - 01	1.296E - 01	1.474E - 01	1.534E - 01	1.540E - 01
	Mean	1.570E - 01	1.582E - 01	1.551E - 01	1.451E - 01	1.529E - 01	1.558E - 01	1.602E - 01
	SD	1.546E - 03	1.042E - 03	3.380E - 03	4.554E - 03	2.718E - 03	1.065E - 03	1.703E - 03
WFG9	Max	2.308E - 01	2.310E - 01	2.311E - 01	2.322E - 01	2.283E - 01	2.313E - 01	2.341E - 01
	Min	2.232E - 01	2.251E - 01	2.227E - 01	2.255E - 01	1.252E - 01	2.263E - 01	1.318E - 01
	Mean	2.276E - 01	2.298E - 01	2.178E - 01	2.290E - 01	2.113E - 01	2.287E - 01	2.285E - 01
	SD	1.840E - 03	1.298E - 03	2.006E - 03	1.548E - 03	3.366E - 02	1.424E - 03	3.450E - 02

TABLE 6: Statistical results of the Times indicator obtained by different algorithms for the WFG problems with 2 objectives.

Functions	Times	Algorithms						
		NSGA-II	NSGA-III	ε -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	3.964E + 00	3.767E + 00	3.557E + 00	3.927E + 00	1.841E + 00	2.352E + 00	3.684E + 00
	Min	2.808E + 00	2.648E + 00	2.492E + 00	2.708E + 00	1.469E + 00	1.741E + 00	3.067E + 00
	Mean	3.357E + 00	3.091E + 00	2.912E + 00	3.205E + 00	1.642E + 00	2.090E + 00	3.325E + 00
	SD	2.148E - 01	2.625E - 01	2.287E - 01	2.992E - 01	9.585E - 02	1.712E - 01	1.466E - 01
WFG2	Max	1.071E + 00	1.419E + 00	1.329E + 00	5.720E - 01	7.134E - 01	6.820E - 01	6.424E - 01
	Min	8.926E - 01	1.130E + 00	9.577E - 01	3.351E - 01	3.878E - 01	3.594E - 01	4.720E - 01
	Mean	9.686E - 01	1.269E + 00	1.114E + 00	4.264E - 01	5.229E - 01	5.039E - 01	5.356E - 01
	SD	4.895E - 02	6.029E - 02	1.029E - 01	5.993E - 02	7.807E - 02	7.390E - 02	4.307E - 02
WFG3	Max	4.544E + 00	4.503E + 00	8.725E + 00	4.700E + 00	8.897E + 00	4.276E + 00	4.383E + 00
	Min	4.025E + 00	4.255E + 00	5.856E + 00	4.275E + 00	6.634E + 00	3.836E + 00	3.410E + 00
	Mean	4.165E + 00	4.356E + 00	7.388E + 00	4.491E + 00	7.981E + 00	4.037E + 00	3.724E + 00
	SD	1.304E - 01	6.370E - 02	5.633E - 01	9.786E - 02	5.795E - 01	1.311E - 01	1.934E - 01
WFG4	Max	6.624E + 00	7.652E + 00	1.014E + 01	6.290E + 00	6.220E + 00	7.120E + 00	4.619E + 00
	Min	6.378E + 00	6.721E + 00	8.649E + 00	4.976E + 00	5.096E + 00	6.195E + 00	3.838E + 00
	Mean	6.505E + 00	6.922E + 00	9.460E + 00	5.694E + 00	5.662E + 00	6.534E + 00	4.238E + 00
	SD	5.756E - 02	1.607E - 01	3.780E - 01	2.880E - 01	2.855E - 01	2.146E - 01	1.624E - 01
WFG5	Max	4.160E + 00	4.619E + 00	8.880E + 00	7.595E + 00	1.173E + 01	1.526E + 01	3.993E + 00
	Min	3.981E + 00	4.462E + 00	7.085E + 00	3.816E + 00	6.699E + 00	1.204E + 01	3.353E + 00
	Mean	4.093E + 00	4.536E + 00	8.023E + 00	4.863E + 00	8.559E + 00	1.349E + 01	3.773E + 00
	SD	3.828E - 02	4.024E - 02	4.490E - 01	8.183E - 01	1.287E + 00	7.334E - 01	1.685E - 01
WFG6	Max	2.158E + 00	2.449E + 00	4.091E + 00	3.982E + 00	7.980E + 00	1.898E + 00	1.675E + 00
	Min	1.836E + 00	2.139E + 00	3.061E + 00	2.397E + 00	2.307E + 00	1.304E + 00	1.347E + 00
	Mean	2.092E + 00	2.336E + 00	3.618E + 00	3.151E + 00	3.688E + 00	1.608E + 00	1.528E + 00
	SD	6.471E - 02	7.171E - 02	2.369E - 01	3.580E - 01	1.083E + 00	1.491E - 01	9.100E - 02
WFG7	Max	1.169E + 01	1.212E + 01	2.167E + 01	1.209E + 01	1.705E + 01	2.071E + 01	1.022E + 01
	Min	1.138E + 01	1.183E + 01	1.976E + 01	1.123E + 01	1.499E + 01	1.913E + 01	9.209E + 00
	Mean	<i>1.153E + 01</i>	1.194E + 01	2.067E + 01	1.155E + 01	1.596E + 01	1.991E + 01	9.607E + 00
	SD	8.228E - 02	7.887E - 02	4.601E - 01	2.109E - 01	5.542E - 01	4.087E - 01	2.124E - 01
WFG8	Max	2.152E + 00	2.392E + 00	3.237E + 00	2.500E + 00	4.035E + 00	1.925E + 00	1.623E + 00
	Min	1.984E + 00	2.226E + 00	2.458E + 00	2.187E + 00	2.925E + 00	1.536E + 00	1.207E + 00
	Mean	2.060E + 00	2.319E + 00	2.919E + 00	2.332E + 00	3.541E + 00	1.742E + 00	1.373E + 00
	SD	3.506E - 02	4.142E - 02	1.622E - 01	8.102E - 02	2.800E - 01	9.259E - 02	9.533E - 02
WFG9	Max	1.333E + 01	1.396E + 01	3.030E + 01	1.329E + 01	1.798E + 01	1.766E + 01	1.366E + 01
	Min	1.149E + 01	1.206E + 01	1.811E + 01	1.091E + 01	1.392E + 01	1.499E + 01	9.763E + 00
	Mean	1.218E + 01	1.266E + 01	2.173E + 01	<i>1.213E + 01</i>	1.632E + 01	1.640E + 01	1.075E + 01
	SD	5.191E - 01	5.213E - 01	3.400E + 00	5.491E - 01	1.058E + 00	6.509E - 01	1.113E + 00

TABLE 7: Statistical results of the Δ_p indicator obtained by different algorithms for the WFG problems with 3 objectives.

Functions	Δ_p	Algorithms						
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	5.857E-01	6.439E-01	6.590E-01	6.421E-01	5.322E-01	6.006E-01	7.057E-01
	Min	5.075E-01	5.770E-01	6.170E-01	6.097E-01	5.021E-01	5.523E-01	6.362E-01
	Mean	5.637E-01	6.081E-01	6.440E-01	6.209E-01	5.169E-01	5.828E-01	6.688E-01
	SD	1.816E-02	1.526E-02	9.889E-03	7.808E-03	6.408E-03	1.421E-02	1.708E-02
WFG2	Max	7.913E-02	6.738E-02	7.615E-02	1.136E-01	8.929E-02	9.255E-02	4.713E-02
	Min	3.831E-02	4.266E-02	4.438E-02	7.703E-02	5.375E-02	3.868E-02	2.953E-02
	Mean	5.027E-02	4.897E-02	6.064E-02	9.996E-02	6.838E-02	5.455E-02	3.660E-02
	SD	9.146E-03	5.690E-03	8.559E-03	1.028E-02	9.356E-03	1.435E-02	4.399E-03
WFG3	Max	8.240E-01	8.902E-01	1.029E+00	8.125E-01	9.034E-01	8.769E-01	8.187E-01
	Min	6.256E-01	6.736E-01	9.144E-01	7.240E-01	7.418E-01	7.769E-01	6.263E-01
	Mean	7.121E-01	8.144E-01	9.725E-01	7.594E-01	8.125E-01	8.281E-01	7.287E-01
	SD	4.258E-02	5.841E-02	2.667E-02	2.247E-02	3.910E-02	2.367E-02	5.544E-02
WFG4	Max	8.911E-02	5.596E-02	5.703E-02	8.811E-02	1.343E-01	6.620E-02	4.899E-02
	Min	6.440E-02	5.174E-02	5.025E-02	7.006E-02	8.485E-02	5.248E-02	4.149E-02
	Mean	7.124E-02	5.370E-02	5.391E-02	7.700E-02	1.011E-01	5.989E-02	4.449E-02
	SD	5.018E-03	8.658E-04	1.859E-03	4.868E-03	1.431E-02	3.269E-03	1.891E-03
WFG5	Max	8.229E-02	6.334E-02	4.445E-02	8.983E-02	9.643E-02	6.710E-02	6.187E-02
	Min	7.097E-02	5.990E-02	3.336E-02	7.453E-02	7.867E-02	6.120E-02	5.241E-02
	Mean	7.648E-02	6.078E-02	3.776E-02	8.087E-02	8.764E-02	6.440E-02	5.602E-02
	SD	2.934E-03	7.096E-04	2.294E-03	3.419E-03	3.822E-03	1.521E-03	2.535E-03
WFG6	Max	1.025E-01	7.740E-02	9.569E-02	1.026E-01	1.362E-01	5.626E-02	8.767E-02
	Min	7.821E-02	6.200E-02	5.607E-02	8.407E-02	9.013E-02	1.846E-02	6.181E-02
	Mean	8.971E-02	6.938E-02	8.187E-02	9.191E-02	1.061E-01	3.097E-02	7.806E-02
	SD	6.079E-03	3.611E-03	9.798E-03	4.422E-03	1.071E-02	8.657E-03	6.414E-03
WFG7	Max	7.608E-02	6.288E-02	6.498E-02	1.114E-01	9.321E-02	7.777E-02	4.037E-02
	Min	6.212E-02	5.938E-02	4.303E-02	9.954E-02	7.654E-02	6.489E-02	3.157E-02
	Mean	6.994E-02	6.072E-02	5.250E-02	1.044E-01	8.441E-02	7.151E-02	3.528E-02
	SD	3.062E-03	9.551E-04	5.056E-03	3.509E-03	3.443E-03	3.066E-03	2.154E-03
WFG8	Max	5.616E-02	1.027E-01	1.348E-01	1.869E-01	1.360E-01	1.034E-01	9.651E-02
	Min	4.580E-02	8.838E-02	1.069E-01	1.396E-01	1.159E-01	9.411E-02	7.971E-02
	Mean	4.946E-02	9.547E-02	1.204E-01	1.602E-01	1.255E-01	9.782E-02	8.843E-02
	SD	2.577E-03	3.067E-03	7.078E-03	1.201E-02	5.490E-03	2.444E-03	4.811E-03
WFG9	Max	1.040E-01	6.317E-02	4.701E-02	8.762E-02	1.204E-01	1.018E-01	5.071E-02
	Min	6.708E-02	5.583E-02	3.034E-02	6.887E-02	7.850E-02	6.713E-02	2.197E-02
	Mean	7.333E-02	5.873E-02	3.676E-02	7.640E-02	9.638E-02	7.550E-02	3.098E-02
	SD	6.754E-03	1.407E-03	3.956E-03	3.594E-03	1.000E-02	9.646E-03	6.226E-03

TABLE 8: Statistical results of the SP indicator obtained by different algorithms for the WFG problems with 3 objectives.

Functions	SP	Algorithms						
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	3.085E - 01	8.545E - 02	4.103E - 02	6.136E - 01	5.475E - 01	6.259E - 01	5.051E - 01
	Min	5.835E - 02	5.663E - 02	2.359E - 02	5.281E - 01	4.439E - 01	5.833E - 01	3.981E - 01
	Mean	8.484E - 02	6.846E - 02	3.000E - 02	5.941E - 01	5.031E - 01	5.998E - 01	4.409E - 01
	SD	4.410E - 02	7.417E - 03	4.651E - 03	1.582E - 02	2.934E - 02	1.076E - 02	2.246E - 02
WFG2	Max	2.708E - 01	1.917E - 01	1.023E - 01	3.927E - 01	4.686E - 01	3.572E - 01	2.003E - 01
	Min	1.454E - 01	1.296E - 01	5.854E - 02	1.400E - 01	1.142E - 01	1.391E - 01	9.928E - 02
	Mean	1.907E - 01	1.593E - 01	7.093E - 02	2.186E - 01	1.990E - 01	2.084E - 01	1.406E - 01
	SD	3.311E - 02	1.540E - 02	1.079E - 02	6.788E - 02	8.695E - 02	5.875E - 02	2.625E - 02
WFG3	Max	1.526E - 01	1.721E - 01	4.324E - 02	1.585E - 01	2.024E - 01	1.494E - 01	4.979E - 02
	Min	9.274E - 02	9.738E - 02	3.118E - 02	1.038E - 01	1.464E - 01	9.174E - 02	3.839E - 02
	Mean	1.168E - 01	1.267E - 01	3.668E - 02	1.320E - 01	1.738E - 01	1.236E - 01	4.222E - 02
	SD	1.313E - 02	1.827E - 02	2.593E - 03	1.273E - 02	1.451E - 02	1.462E - 02	2.523E - 03
WFG4	Max	2.560E - 01	2.469E - 01	5.580E - 02	2.942E - 01	4.067E - 01	2.727E - 01	8.951E - 02
	Min	1.787E - 01	1.974E - 01	4.728E - 02	1.901E - 01	2.618E - 01	1.941E - 01	5.761E - 02
	Mean	2.099E - 01	2.236E - 01	5.002E - 02	2.333E - 01	3.179E - 01	2.248E - 01	6.549E - 02
	SD	2.243E - 02	1.125E - 02	2.149E - 03	2.309E - 02	2.946E - 02	2.010E - 02	6.137E - 03
WFG5	Max	2.791E - 01	2.600E - 01	5.639E - 02	2.443E - 01	3.383E - 01	2.225E - 01	9.423E - 02
	Min	1.671E - 01	2.073E - 01	4.536E - 02	1.729E - 01	2.644E - 01	1.678E - 01	5.678E - 02
	Mean	2.136E - 01	2.313E - 01	5.034E - 02	2.001E - 01	2.998E - 01	1.920E - 01	7.442E - 02
	SD	2.163E - 02	1.355E - 02	2.364E - 03	1.876E - 02	2.037E - 02	1.600E - 02	9.141E - 03
WFG6	Max	2.782E - 01	2.547E - 01	6.708E - 02	2.605E - 01	3.582E - 01	2.519E - 01	1.034E - 01
	Min	1.675E - 01	2.123E - 01	5.684E - 02	1.582E - 01	2.657E - 01	1.631E - 01	7.831E - 02
	Mean	2.176E - 01	2.265E - 01	6.228E - 02	2.083E - 01	3.128E - 01	2.080E - 01	8.773E - 02
	SD	2.555E - 02	1.008E - 02	3.131E - 03	2.524E - 02	2.627E - 02	2.798E - 02	6.767E - 03
WFG7	Max	2.704E - 01	2.640E - 01	5.372E - 02	2.963E - 01	3.660E - 01	3.063E - 01	7.872E - 02
	Min	1.967E - 01	2.201E - 01	4.488E - 02	1.540E - 01	2.533E - 01	2.216E - 01	5.895E - 02
	Mean	2.267E - 01	2.383E - 01	4.877E - 02	2.055E - 01	3.069E - 01	2.499E - 01	6.723E - 02
	SD	2.023E - 02	1.223E - 02	2.382E - 03	2.483E - 02	3.077E - 02	2.142E - 02	4.472E - 03
WFG8	Max	2.633E - 01	2.943E - 01	6.977E - 02	2.517E - 01	3.675E - 01	2.676E - 01	9.905E - 02
	Min	1.941E - 01	1.941E - 01	5.645E - 02	1.729E - 01	2.549E - 01	1.588E - 01	6.202E - 02
	Mean	2.276E - 01	2.493E - 01	6.401E - 02	2.067E - 01	3.034E - 01	2.196E - 01	7.033E - 02
	SD	1.842E - 02	2.578E - 02	3.342E - 03	1.658E - 02	2.765E - 02	2.396E - 02	6.679E - 03
WFG9	Max	2.485E - 01	2.504E - 01	5.528E - 02	2.419E - 01	3.724E - 01	2.210E - 01	6.936E - 02
	Min	1.749E - 01	2.003E - 01	3.768E - 02	1.616E - 01	2.634E - 01	1.574E - 01	4.894E - 02
	Mean	2.079E - 01	2.258E - 01	4.560E - 02	1.938E - 01	3.084E - 01	2.021E - 01	5.640E - 02
	SD	1.865E - 02	1.214E - 02	3.975E - 03	1.819E - 02	2.408E - 02	1.548E - 02	4.673E - 03

TABLE 9: Statistical results of the HV indicator obtained by different algorithms for the WFG problems with 3 objectives.

Functions	HV	Algorithms					RMOABC
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3
WFG1	Max	1.562E - 01	2.069E - 01	7.142E - 02	7.035E - 02	2.499E - 01	7.737E - 02
	Min	1.127E - 01	1.568E - 01	0.000E + 00	9.377E - 04	2.200E - 01	9.640E - 03
	Mean	1.313E - 01	1.841E - 01	1.748E - 02	4.181E - 02	2.339E - 01	4.650E - 02
	SD	1.133E - 02	1.323E - 02	1.454E - 02	1.927E - 02	7.751E - 03	2.048E - 02
WFG2	Max	8.899E - 01	9.022E - 01	8.870E - 01	8.267E - 01	8.835E - 01	8.931E - 01
	Min	8.691E - 01	8.674E - 01	8.511E - 01	7.884E - 01	8.468E - 01	8.776E - 01
	Mean	8.807E - 01	8.912E - 01	8.698E - 01	8.094E - 01	8.671E - 01	8.860E - 01
	SD	6.045E - 03	6.742E - 03	1.013E - 02	1.081E - 02	8.704E - 03	4.321E - 03
WFG3	Max	2.996E - 01	2.812E - 01	2.770E - 01	2.603E - 01	2.183E - 01	2.677E - 01
	Min	2.546E - 01	2.534E - 01	2.298E - 01	1.894E - 01	1.285E - 01	2.268E - 01
	Mean	2.830E - 01	2.662E - 01	2.550E - 01	2.233E - 01	1.657E - 01	2.455E - 01
	SD	1.092E - 02	7.071E - 03	1.312E - 02	1.805E - 02	1.857E - 02	8.239E - 03
WFG4	Max	3.564E - 01	3.877E - 01	3.797E - 01	3.961E - 01	3.558E - 01	3.657E - 01
	Min	3.286E - 01	3.741E - 01	3.598E - 01	3.823E - 01	3.150E - 01	3.475E - 01
	Mean	3.394E - 01	3.823E - 01	3.705E - 01	3.897E - 01	3.327E - 01	3.581E - 01
	SD	6.287E - 03	3.306E - 03	4.770E - 03	3.306E - 03	9.216E - 03	5.227E - 03
WFG5	Max	3.351E - 01	3.729E - 01	4.002E - 01	3.328E - 01	3.254E - 01	3.629E - 01
	Min	3.112E - 01	3.566E - 01	3.823E - 01	3.009E - 01	3.003E - 01	3.436E - 01
	Mean	3.248E - 01	3.676E - 01	3.935E - 01	3.179E - 01	3.131E - 01	3.540E - 01
	SD	6.226E - 03	3.928E - 03	4.333E - 03	9.265E - 03	6.171E - 03	4.625E - 03
WFG6	Max	3.268E - 01	3.595E - 01	3.921E - 01	3.243E - 01	3.314E - 01	3.406E - 01
	Min	2.509E - 01	2.972E - 01	2.980E - 01	2.864E - 01	2.652E - 01	2.740E - 01
	Mean	2.901E - 01	3.327E - 01	3.236E - 01	3.047E - 01	2.956E - 01	3.086E - 01
	SD	1.621E - 02	1.656E - 02	1.966E - 02	8.537E - 03	1.653E - 02	1.446E - 02
WFG7	Max	3.539E - 01	3.787E - 01	3.843E - 01	2.807E - 01	3.444E - 01	3.531E - 01
	Min	3.347E - 01	3.583E - 01	3.561E - 01	2.416E - 01	3.069E - 01	3.270E - 01
	Mean	3.461E - 01	3.722E - 01	3.708E - 01	2.581E - 01	3.255E - 01	3.380E - 01
	SD	5.267E - 03	5.421E - 03	7.595E - 03	8.607E - 03	8.995E - 03	5.934E - 03
WFG8	Max	2.854E - 01	3.056E - 01	2.999E - 01	2.177E - 01	2.653E - 01	3.006E - 01
	Min	2.633E - 01	2.836E - 01	2.674E - 01	1.825E - 01	2.361E - 01	2.816E - 01
	Mean	2.739E - 01	2.936E - 01	2.818E - 01	1.980E - 01	2.499E - 01	2.910E - 01
	SD	5.439E - 03	4.792E - 03	7.947E - 03	8.320E - 03	7.743E - 03	4.693E - 03
WFG9	Max	3.474E - 01	3.746E - 01	4.029E - 01	3.281E - 01	3.396E - 01	3.529E - 01
	Min	3.029E - 01	3.379E - 01	3.702E - 01	2.968E - 01	2.130E - 01	2.491E - 01
	Mean	3.318E - 01	3.598E - 01	3.884E - 01	3.137E - 01	3.113E - 01	3.296E - 01
	SD	1.174E - 02	6.881E - 03	8.606E - 03	8.395E - 03	2.881E - 02	1.767E - 02

TABLE 10: Statistical results of the Times indicator obtained by different algorithms for the WFG problems with 3 objectives.

Functions	Times	Algorithms						
		NSGA-II	NSGA-III	ϵ -MOEA	SMPSO	MOEA/D	GDE3	RMOABC
WFG1	Max	1.714E + 01	2.755E + 01	4.232E + 01	1.686E + 01	1.061E + 01	1.648E + 01	7.372E + 00
	Min	1.321E + 01	1.963E + 01	3.007E + 01	1.474E + 01	8.835E + 00	1.415E + 01	6.949E + 00
	Mean	1.405E + 01	2.408E + 01	3.723E + 01	1.591E + 01	9.773E + 00	1.524E + 01	7.227E + 00
	SD	1.180E + 00	2.032E + 00	3.331E + 00	4.368E - 01	3.830E - 01	5.030E - 01	9.741E - 02
WFG2	Max	2.088E + 01	4.028E + 01	1.032E + 02	2.160E + 01	1.403E + 01	2.099E + 01	1.035E + 01
	Min	1.816E + 01	2.802E + 01	7.111E + 01	1.830E + 01	1.219E + 01	1.815E + 01	9.545E + 00
	Mean	1.877E + 01	3.395E + 01	8.728E + 01	1.956E + 01	1.319E + 01	1.956E + 01	9.747E + 00
	SD	9.029E - 01	2.717E + 00	9.289E + 00	8.509E - 01	5.038E - 01	9.577E - 01	1.502E - 01
WFG3	Max	2.568E + 00	4.149E + 01	2.535E + 02	7.538E + 00	1.703E + 00	1.353E + 01	2.369E + 00
	Min	2.002E + 00	3.061E + 01	1.717E + 02	5.360E + 00	1.530E + 00	1.231E + 01	1.820E + 00
	Mean	2.179E + 00	3.496E + 01	2.124E + 02	6.428E + 00	1.597E + 00	1.287E + 01	2.006E + 00
	SD	1.789E - 01	2.422E + 00	1.673E + 01	5.315E - 01	4.055E - 02	3.292E - 01	1.653E - 01
WFG4	Max	7.610E + 01	6.136E + 02	1.013E + 03	7.815E + 01	4.033E + 01	6.893E + 01	4.030E + 01
	Min	7.086E + 01	3.675E + 02	7.970E + 02	7.291E + 01	3.593E + 01	6.604E + 01	3.468E + 01
	Mean	7.251E + 01	4.956E + 02	9.068E + 02	7.569E + 01	3.872E + 01	6.660E + 01	3.738E + 01
	SD	1.297E + 00	4.899E + 01	5.167E + 01	1.372E + 00	1.245E + 00	5.100E - 01	1.538E + 00
WFG5	Max	6.931E + 01	4.090E + 02	8.132E + 02	5.413E + 02	4.979E + 01	7.366E + 01	3.834E + 01
	Min	6.487E + 01	2.423E + 02	6.025E + 02	6.403E + 01	4.613E + 01	7.022E + 01	3.257E + 01
	Mean	6.702E + 01	3.022E + 02	6.978E + 02	8.082E + 01	4.800E + 01	7.246E + 01	3.493E + 01
	SD	1.193E + 00	4.391E + 01	5.262E + 01	8.697E + 01	1.025E + 00	9.243E - 01	1.427E + 00
WFG6	Max	6.395E + 01	3.368E + 02	1.855E + 04	6.681E + 01	3.925E + 01	6.572E + 01	3.243E + 01
	Min	6.299E + 01	2.357E + 02	3.978E + 02	6.499E + 01	3.342E + 01	6.437E + 01	3.204E + 01
	Mean	6.346E + 01	2.770E + 02	1.081E + 03	6.609E + 01	3.640E + 01	6.496E + 01	3.230E + 01
	SD	1.917E - 01	2.562E + 01	3.301E + 03	4.290E - 01	1.401E + 00	3.224E - 01	8.710E - 02
WFG7	Max	6.070E + 01	4.729E + 02	8.191E + 02	6.678E + 01	4.132E + 01	8.029E + 01	3.448E + 01
	Min	5.898E + 01	3.627E + 02	6.596E + 02	6.384E + 01	3.562E + 01	7.275E + 01	3.027E + 01
	Mean	5.969E + 01	3.981E + 02	7.463E + 02	6.526E + 01	3.875E + 01	7.508E + 01	3.120E + 01
	SD	4.673E - 01	2.470E + 01	4.053E + 01	7.469E - 01	1.268E + 00	1.365E + 00	8.456E - 01
WFG8	Max	6.678E + 01	3.988E + 03	5.919E + 02	7.215E + 01	3.747E + 01	6.681E + 01	3.516E + 01
	Min	6.412E + 01	3.738E + 02	4.801E + 02	6.895E + 01	3.022E + 01	6.565E + 01	3.229E + 01
	Mean	6.546E + 01	5.427E + 02	5.325E + 02	7.034E + 01	3.447E + 01	6.643E + 01	3.364E + 01
	SD	6.233E - 01	6.511E + 02	2.912E + 01	8.833E - 01	1.722E + 00	2.367E - 01	8.721E - 01
WFG9	Max	6.706E + 01	5.353E + 02	2.001E + 03	6.971E + 01	5.869E + 01	8.247E + 01	3.500E + 01
	Min	6.431E + 01	2.799E + 02	6.042E + 02	6.671E + 01	4.682E + 01	7.644E + 01	3.307E + 01
	Mean	6.588E + 01	4.194E+02	8.709E + 02	6.791E + 01	5.213E + 01	7.941E + 01	3.405E + 01
	SD	6.416E - 01	6.430E + 01	2.321E + 02	6.776E - 01	2.355E + 00	1.452E + 00	6.175E - 01

FIGURE 6: Normalized ranges of the five objective functions' values of the water problem obtained by the seven algorithms.

value-high value [70]. Thus, the maximum and minimum values of the normalized objectives of Table 11 are presented in Figure 6. The lines indicate the coverage of the objective values for the nondominated solution set obtained by the seven multiobjectives algorithms.

It can be concluded from the figure, the seven algorithms perform better in the objective function f_2 , f_3 , and f_5 which most of the value space can be covered. But for the objective function f_1 , all the seven algorithms can not obtain the better coverage, which they cover the range from 0.8 to 1.0 of the value space. And for the objective function f_4 , the NSGA-II, NSGA-III, ε -MOEA, and SMPSO obtained much larger values than the Pareto front. MOEA/D and GDE3, and especially RMOABC, got better coverage for this objective function.

6.3.2. Optimization Process. Because the WRP problem is a many-objectives (5 objectives) issue and is very complicated, the calculation results of the distribution indicator (spacing) are very high and do not have too many significant representatives. Thus, to compare the performance of the seven multiobjective algorithms for this problem, we took the performance indicator HV as the quality measure in this subsection. The variations of these two indicators over the iteration number of the seven algorithms are shown in Figure 7. The X axis is the iteration number of the algorithms, and the Y axis represents the values HV of the solutions set of the different algorithms, respectively.

It can be seen from the figures, for the HV indicator, ε -MOEA and RMOABC perform better than other algorithms, especially the RMOABC algorithm can exceed the others at about 900 iterations. The other algorithms perform poor, and SMPSO and NSGA-II have a fluctuating line of smaller values.

FIGURE 7: The performance indicator of HV vs. the iteration number of the seven multiobjective algorithms for WRP problem.

6.3.3. Multiobjective Performance Comparison. Through the thirty independent runs of the seven algorithms, the performance metric of HV and execution time for the WRP problem are shown in Table 12. The best mean among the algorithms for each WFG function is shown in bold and italics, respectively.

For the HV indicators of the WRP problem, RMOABC obtained the best value, and ε -MOEA obtained the second best value. It demonstrated that RMOABC could exceed other algorithms in the performance of convergence and quality for the WRP problem. For the execution times

TABLE 11: Values ranges of five objective functions for water resources planning problem obtained by the seven algorithms.

Objective Function	Value Range	Algorithms						
		NSGA-II	NSGA-III	ε -MOEA	SMP SO	MOEA/D	GDE3	RMOABC
f_1	Max	78758.58	76429.01	78587.93	83060.74	79119.91	74160.93	79991.59
	Min	63840.31	63842.71	63849.61	63840.28	63840.28	63840.28	63840.28
f_2	Max	1350.00	1349.97	1349.90	1350.00	1350.00	1350.00	1350.00
	Min	41.27	43.74	41.35	43.23	49.97	41.35	41.05
f_3	Max	2853468.42	2852845.28	2853441.68	2853468.96	2853468.96	2853468.96	2853468.96
	Min	285346.90	285392.12	285378.67	285346.90	285346.90	285346.90	285346.90
f_4	Max	10062944.88	11435067.73	11914875.77	16027735.33	6599674.37	7015174.18	5900700.76
	Min	183771.00	184068.53	183942.39	183749.97	183749.97	183749.97	183749.97
f_5	Max	24986.12	24997.07	24997.74	24993.31	24849.60	24989.64	24985.27
	Min	9.26	75.68	45.33	7.22	7.22	7.22	7.22

TABLE 12: Performance comparison of the seven algorithms for water resource plan (WRP) problem.

Indicators		Algorithms						
		NSGA-II	NSGA-III	ε -MOEA	SMP SO	MOEA/D	GDE3	RMOABC
HV	Max	4.018E - 01	4.314E - 01	4.534E - 01	3.914E - 01	4.148E - 01	4.332E - 01	4.599E - 01
	Min	3.470E - 01	4.070E - 01	4.388E - 01	2.515E - 01	2.638E - 01	4.259E - 01	4.336E - 01
	Mean	3.864E - 01	4.213E - 01	4.451E - 01	3.346E - 01	3.776E - 01	4.301E - 01	4.534E - 01
	SD	1.137E - 02	4.885E - 03	3.756E - 03	3.211E - 02	3.887E - 02	1.887E - 03	4.832E - 03
Times	Max	8.137E + 00	1.257E + 01	1.632E + 02	1.098E + 01	6.698E + 00	1.443E + 01	7.555E + 01
	Min	7.514E + 00	1.049E + 01	9.670E + 01	9.733E + 00	5.790E + 00	1.268E + 01	2.273E + 01
	Mean	7.774E + 00	1.115E + 01	1.231E + 02	1.034E + 01	6.206E + 00	1.340E + 01	4.621E + 01
	SD	1.505E - 01	4.065E - 01	1.553E + 01	3.475E - 01	2.411E - 01	3.916E - 01	1.221E + 01

indicator, MOEA/D has the best value, and NSGA-II has the second best value; RMOABC and ε -MOEA have to take more time to obtain the final solution set.

7. Conclusions

In this paper, we have presented a unified, flexible, configurable, and user-friendly MOABC algorithm framework, UOF-MOABC, which combines a multiobjective ABC algorithm named RMOABC and the multiobjective evolution algorithms (MOEA) framework. Particularly, the core classes of the MOEA framework are described. How to implement the new RMOABC algorithm into the framework has also been illustrated. Then, the Walking Fish Group (WFG) test suite and a many-objective water resource planning (WRP) problem were taken and compared with other six state-of-the-art multiobjective algorithms (NSGA-II, NSGA-III, ε -MOEA, SMP SO, MOEA/D, and GDE3) for verification and application.

The obtained experiment results have been statistically analyzed on the basis of three quality indicators (Δ_p , SP, and HV) with the computation time under the framework. It can be drawn from the experiments that according to the problems, parameter settings, and quality indicators used, the RMOABC method generally outperforms or is equal to the other six MOO algorithms in our study.

As for future work, we plan to integrate more realistic problems to provide more reliable nondominated solution sets for supporting decision making, as well as considering the parallelization of multiobjective ABC algorithms to improve the efficiency and accuracy for solving complex MOPs.

Data Availability

The Pareto fronts data used to support the findings of this study are available from the project of jMetal 5.0 which can be downloaded from the web site: <https://jmetal.github.io/jMetal/>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (Grant nos. 61462058 and 61862038).

References

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2001.
- [2] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, Vol. 135, Springer Science and Business Media, Berlin, Germany, 2001.
- [3] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
- [4] C. Coello, G. Lamont, and D. van Veldhuizen, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2007.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE*

- Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [6] K. Deb, M. Mohan, and S. Mishra, “A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions,” KanGAL Report No 2003002, 2003.
 - [7] D. Hadka and P. R. Borg, “An auto-adaptive many-objective evolutionary computing framework,” *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.
 - [8] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, San Fransisco, CA, USA, 2001.
 - [9] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November–December 1995.
 - [10] M. Dorigo and M. Birattari, *Ant Colony Optimization*, *Encyclopedia of Machine Learning*, Springer, Boston, MA, USA, 2011.
 - [11] M. Eusuff, K. Lansey, and F. Pasha, “Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization,” *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
 - [12] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical Report-TR06, Erciyes University, Kayseri, Turkey, 2005.
 - [13] C. S. Zhang, *Research and Application of Multi-Objective Artificial Bee Colony Algorithm and Genetic Algorithm*, Vol. 7, Northeastern University Press, Shenyang, China, 2013.
 - [14] B. Christian, P. Jakob, R. R. Günther, and R. Andrea, “Hybrid metaheuristics in combinatorial optimization: a survey,” *Applied Soft Computing*, vol. 11, no. 6, pp. 4135–4151, 2011.
 - [15] J. Durillo and A. Nebro, “jMetal: a java framework for multi-objective optimization,” *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.
 - [16] A. Liefvooghe, M. Basseur, L. Jourdan, and E.-G. Talbi, “ParadisEO-MOEO: a framework for evolutionary multi-objective optimization,” in *Fourth International Conference on Evolutionary Multi-criterion Optimization (EMO 2007)*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4403 of LNCS, pp. 386–400, Springer, Matsushima, Japan, March 2007.
 - [17] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, “PISA—a platform and programming language independent interface for search algorithms,” in *Evolutionary Multi-criterion Optimization (EMO 2003)*, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds., Lecture Notes in Computer Science, pp. 494–508, Springer, Faro, Portugal, April 2003.
 - [18] D. Hadka, “MOEA framework—a free and open source java framework for multiobjective optimization, version 2.12,” 2017, <http://www.moeaframework.org/>.
 - [19] J. Y. Huo and L. Q. Liu, “An improved multi-objective artificial bee colony optimization algorithm with regulation operators,” *Information*, vol. 8, no. 1, p. 18, 2017.
 - [20] C. Y. Lin and J. L. Lin, *Method and Theory of Multi-objective Optimization*, Jilin Education Press, Changchun, China, 1992.
 - [21] J. H. Zheng, *Multi-Objective Evolutionary Algorithm and its Applications*, Science Press, Beijing, China, 2007.
 - [22] J. D. Schaffer, “Multiple objective optimization with vector evaluated genetic algorithms,” in *Proceedings of 1st international Conference on Genetic Algorithms*, L. Erlbaum Associates Inc., pp. 93–100, Pittsburgh, PA, USA, July 1985.
 - [23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional location, Boston, MA, USA, 1989.
 - [24] X. X. Cui, *Multi-Objective Evolutionary Algorithm and Its Applications*, National Defense Industry Press, Beijing, China, 2008.
 - [25] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated pareto sets,” *MOEA/D and NSGA-II*, vol. 13, no. 2, pp. 184–302, 2009.
 - [26] C. R. Raquel and P. C. Naval, “An effective use of crowding distance in multiobjective particle swarm optimization,” in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2005)*, pp. 257–264, Washington, DC, USA, June 2005.
 - [27] C. S. Tsou, S. C. Chang, and P. W. Lai, “Using crowding distance to improve multi-objective PSO with local search,” in *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, F. T. S. Chan and M. K. Tiwari, Eds., Itech Education and Publishing, Vienna, Austria, 2007.
 - [28] S. V. Kamble, S. U. Mane, and A. J. Umbarkar, “Hybrid multi-objective particle swarm optimization for flexible job shop scheduling problem,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 7, no. 4, pp. 54–61, 2015.
 - [29] W. F. Leong and G. G. Yen, “PSO-based multiobjective optimization with dynamic population size and adaptive local archives,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 38, p. 5, 2008.
 - [30] J. Y. Huo, Y. N. Zhang, and H. X. Zhao, “An improved artificial bee colony algorithm for numerical functions,” *International Journal of Reasoning-based Intelligent Systems*, vol. 7, no. 3-4, pp. 200–208, 2015.
 - [31] R. Hedayatzadeh, B. Hasanizadeh, and R. Akbari R, “A multi-objective artificial bee colony for optimization multi-objective problems,” in *Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 277–281, Chengdu, China, August 2010.
 - [32] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, “A multi-objective artificial bee colony algorithm,” *Swarm and Evolutionary Computation*, vol. 2, no. 1, pp. 39–52, 2012.
 - [33] W. P. Zou, Y. L. Zhu, H. N. Chen, and B. W. Zhang, “Solving multi-objective optimization problems using artificial bee colony algorithm,” *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 569784, 37 pages, 2011.
 - [34] R. Akbari and K. Ziarati, “Multi-objective bee swarm optimization,” *International Journal of Innovative Computing Information and Control*, vol. 8, no. 1B, pp. 715–726, 2012.
 - [35] H. Zhang, Y. Zhu, W. Zou, and X. Yan, “A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production,” *Applied Mathematical Modeling*, vol. 36, no. 6, pp. 2578–2591, 2012.
 - [36] J. P. Luo, Q. Liu, Y. Yang, X. Li, M. R. Chen, and W. M. Cao, “An artificial bee colony algorithm for multi-objective optimization,” *Applied Soft Computing*, vol. 50, pp. 235–251, 2017.
 - [37] A. Kishor, P. K. Singh, and J. Prakash, “NSABC: non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering,” *Neuro Computing*, vol. 216, pp. 514–533, 2016.
 - [38] S. K. Nseef, S. Abdullah, A. Turkey, and G. Kendall, “An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems,” *Knowledge Based System*, vol. 104, pp. 14–23, 2016.
 - [39] M. Choobineh and S. Mohagheghi, “A multi-objective optimization framework for energy and asset management in an industrial Microgrid,” *Journal of Cleaner Production*, vol. 139, pp. 1326–1338, 2016.

- [40] K. Khalili-Damghani, M. Tavana, and S. Sadi-Nezhad, "An integrated multi-objective framework for solving multi-period project selection problems," *Applied Mathematics and Computation*, vol. 219, no. 6, pp. 3122–3138, 2012.
- [41] K. Vergidis, D. Saxena, and A. Tiwari, "An evolutionary multi-objective framework for business process optimisation," *Applied Soft Computing*, vol. 12, no. 8, pp. 2638–2653, 2012.
- [42] V. M. Charitopoulos and V. Dua, "A unified framework for model-based multi-objective linear process and energy optimisation under uncertainty," *Applied Energy*, vol. 186, pp. 539–548, 2017.
- [43] S. C. Tsai and S. T. Chen, "A simulation-based multi-objective optimization framework: a case study on inventory management," *Omega*, vol. 70, pp. 148–159, 2017.
- [44] M. G. Avci and H. Selim, "A Multi-objective, simulation-based optimization framework for supply chains with premium freights," *Expert Systems with Applications*, vol. 67, pp. 95–106, 2017.
- [45] P. Golding, S. Kapadia, and S. Naylor, "Framework for minimising the impact of regional shocks on global food security using multi-objective ant colony optimisation," *Environmental Modelling and Software*, vol. 95, pp. 303–319, 2017.
- [46] C. P. Newland, H. R. Maier, A. C. Zecchin, J. P. Newman, and H. van Delden, "Multi-objective optimisation framework for calibration of Cellular Automata land-use models," *Environmental Modelling and Software*, vol. 100, pp. 175–200, 2018.
- [47] J. J. Durillo and A. J. Nebro, *jMetal: A Java Framework for Multi-Objective Optimization*, Elsevier Science Ltd., New York, NY, USA, 2011.
- [48] C. Barba-González, J. García-Nieto, A. J. Nebro et al., "jMetalSP: a framework for dynamic multi-objective big data optimization," *Applied Soft Computing*, vol. 69, pp. 737–748, 2017.
- [49] V. Pareto, *The Rise and Fall of the Elites*, Bedminster Press, Totowa, NJ, USA, 1968.
- [50] Y. Huo, Y. Zhuang, J. Gu, and S. Ni, "Elite-guided multi-objective artificial bee colony algorithm," *Applied Soft Computing*, vol. 32, pp. 199–210, 2015.
- [51] G. P. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [52] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evolution Computing*, vol. 8, no. 2, pp. 149–172, 2000.
- [53] Q. D. Wu, L. Wang, and Y. W. Zhang, "Selection of meta-heuristic algorithms based on standardization evaluation system," in *Proceedings of Plenary Talk in the Sixth International Conference on Swarm Intelligence and the Second BRICS Congress on Computational Intelligence (ICSI-CCI'2015)*, Beijing, China, June 2015.
- [54] J. Guo, J. Z. Zhou, Q. Zou, L. X. Song, and Y. C. Zhang, "Study on multi-objective calibration of hydrological model and effect of objective functions combination on optimization results," *Journal of Sichuan University*, vol. 43, no. 6, pp. 58–63, 2011.
- [55] D. A. Van, V. Gary, and B. Lamont, "Multiobjective evolutionary algorithm research: a history and analysis," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 1998.
- [56] K. Deb, M. Mohan, and S. Mishra, "Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.
- [57] O. S. Rudolph, C. Grimme, C. Domínguez-Medina, and H. Trautmann, "Optimal averaged Hausdorff archives for bi-objective problems: theoretical and numerical results," *Computational Optimization and Applications*, vol. 64, no. 2, pp. 589–618, 2016.
- [58] S. A. R. Mohammadi, M. R. Feizi Derakhshi, and R. Akbari, "An adaptive multi-objective artificial bee colony with crowding distance mechanism," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 37, no. E1, pp. 79–92, 2013.
- [59] 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.
- [60] K. Musselman and J. Talavage, "A trade-off cut approach to multiple objective optimization," *Operations Research*, vol. 28, no. 6, pp. 1424–1435, 1980.
- [61] F. Y. Cheng and X. S. Li, "Generalized method for multi-objective engineering optimization," *Engineering Optimisation*, vol. 31, no. 5, pp. 641–661, 1999.
- [62] S. Huband, P. Hingston, and L. Barone, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [63] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artificial Intelligence*, vol. 228, pp. 45–65, 2015.
- [64] 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.
- [65] K. Deb, "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions," KanGAL Report No 2003002, 2003.
- [66] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "SMPISO: a new PSO-based meta-heuristic for multi-objective optimization," in *Proceedings of IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making*, vol. 66–73, Nashville, TN, USA, March–April 2009.
- [67] 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.
- [68] S. Kukkonen and J. Lampinen, "GDE3: the third evolution step of generalized differential evolution," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, pp. 443–450, Edinburgh, UK, September 2005.
- [69] O. Schutze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 504–522, 2012.
- [70] J. Y. Huo and L. Q. Liu, "Application research of multi-objective Artificial Bee Colony optimization algorithm for parameters calibration of hydrological model," *Neural Computing and Applications*, pp. 1–18, 2018.