IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 7, 2019, accepted November 26, 2019, date of publication December 4, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957637

A Novel Hybrid Multi-Objective Particle Swarm Optimization Algorithm With an Adaptive Resource Allocation Strategy

LINGJIE LI, SHUO CHEN, ZHE GONG, QIUZHEN LIN, AND ZHONG MING
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding authors: Qiuzhen Lin (qiuzhlin@szu.edu.cn) and Zhong Ming (mingz@szu.edu.cn)

This work was supported in part by the Shenzhen Technology Plan under Grant JCYJ20170817102218122, Grant JSGG20180507182904693, and Grant GGFW2018021118145859, in part by the National Natural Science Foundation of China under Grant 61876110, Grant 61836005, and Grant 61672358, in part by the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030313338, and in part by the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University.

ABSTRACT Recently, there are a number of particle swarm optimization algorithms (PSOs) proposed for tackling multi-objective optimization problems (MOPs). Most of multi-objective PSOs (MOPSOs) were designed to speed up their convergence, which have been validated when tackling various kinds of MOPs. However, they may face some challenges for tackling some complicated MOPs, such as the UF test problems with complicated Pareto-optimal sets, mainly due to their neglect on the diversity. To solve the above problem, a novel hybrid MOPSO (called HMOPSO-ARA) is suggested in this paper with an adaptive resource allocation strategy, which shows a superior performance over most MOPSOs. Using the decomposition approach in HMOPSO-ARA, MOPs are transferred into a set of subproblems, each of which is accordingly optimized by one particle using a novel velocity update approach with the strengthened search capability. Then, an adaptive resource allocation strategy is employed based on the relevant improvement on the aggregated function, which can reasonably assign the computational resource to the particles according to their performance, so as to accelerate the convergence speed to the true Pareto-optimal front. Moreover, a decomposition-based clonal selection strategy is further used to enhance our performance, where the cloning process is run on the external archive based on the relevant fitness improvement. The experiments validate the superiority of HMOPSO-ARA over four competitive MOPSOs (SMPSO, CMPSO, dMOPSO and AgMOPSO) and four competitive multi-objective evolutionary algorithms (MOEA/D-ARA, MOEA/D-DE MOEA/D-GRA and EF_PD) when tackling thirty-five test problems (DTLZ1-DTLZ9, WFG1-WFG9, UF1-UF10 and F1-F9), in terms of two widely used performance indicators.

INDEX TERMS Multi-objective optimization, particle swarm optimization, resource allocation.

I. INTRODUCTION

In some real-world engineering problems, we often need to solve the optimization problems with several objectives, which are usually conflicted with each other. Generally, these problems are called multi-objective optimization problems (MOPs) [1] and can be modeled by

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\ &\text{subject to } \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ is a decision vector within the search space Ω, and the objective vector includes m objective functions to be optimized in the search space Ω (n and m are respectively the dimensions of decision space and objective space). The final output of MOPs will produce a Pareto-optimal set (***PS***) with its mapping in the objective space called Pareto-optimal front (***PF***), which includes the best trade-off solutions among the objectives. When tackling MOPs, we need to search a set of approximate solutions with good convergence (i.e., they can approximate the true ***PF*** as closely as possible) and with good diversity (i.e., they can cover the true ***PF*** as evenly as possible).

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

Recently, multi-objective evolutionary algorithms (MOEAs) have been presented and shown promising performance in solving different kinds of MOPs [2]–[4]. Based on the selection mechanisms, most of existing MOEAs can be classified into three main categories, i.e., Pareto-based MOEAs [5]–[9], decomposition-based MOEAs [10]–[17], and indicator-based MOEAs [20]–[23].

Pareto-based MOEAs incorporate the concept of Pareto optimality into the evolutionary selection, such as SPEA2 [5] and NSGA-II [6]. The follow-up work has NSGA-II+AD [7] with a novel angle dominance criterion to provide sufficient selection pressure towards the PF, NSGA-III-SE [8] with a new selection-and-elimination operator to identify the reference point with the minimum niche count, and ar-MOEA [9] with a stricter partial order for non-dominated solutions.

Decomposition-based MOEAs solve MOPs by optimizing a set of subproblems on a collaborative manner. MOEA/D [10] is one famous MOEA based on the decomposition framework, in which each individual is associated to optimize one subproblem. Instead of using Pareto domination relation, the relevant aggregated function values are used to run the solution replacement mechanism.

Recently, there are some modified decomposition-based MOEAs to further improve the performance of MOEA/D. For example, a dynamic resource allocation (DRA) strategy was proposed in MOEA/D-DRA [11], where a utility function based on the improvement of aggregated function is employed to decide which subproblem should be given more computation resource in each generation. A generalized resource allocation strategy was designed in MOEA/D-GRA [12] to enhance DRA by using a probability vector of improvement and a diversity-enhanced resource allocation strategy was reported in MOEA/D-IRA [13] to further consider the solution density of each subproblem. Moreover, a collaborative resource allocation strategy was proposed in MOEA/D-CRA [14], which runs the resource allocation based on the contributions to produce the high-quality solutions for each subproblem. Moreover, a novel Tchbycheff decomposition with *lp*-norm constraint on direction vectors and a new unary R_2 indicator were proposed in MOEA/D-2TCHMFI [15], in which the subproblems objective function endowed with clear geometric property. An adversarial decomposition method was developed in MOEA/AD [16] to leverage the complementary characteristics of different subproblem formulations within a single paradigm. An adaptive decomposition-based approach was presented in ADEA [17], which introduces an adaptation mechanism for decomposition and the weight vector.

Indicator-based MOEAs use performance indicators (e.g., R2 [18] and HV [19]) as their density estimator to guide the evolutionary search, such as IBEA [20] and SMS-EMOA [21]. Recently, a novel indicator-based method was presented in DLS-MOEA [22], which designs an enhanced diversification mechanism and a new solution generator based on the external archive, while an enhanced inverted generational distance indicator was reported in AR-MOEA [23] to enhance the versatility of MOEAs on problems with different PF shapes. For a more detailed review of recent MOEAs, please refer to [24].

On the other hand, a number of particle swarm optimization (PSO) algorithms were also proposed for solving MOPs [25], [26], mainly due to their easy implementation with a fast convergence speed. Initially, PSO was designed to solve single-objective optimization problems (SOPs). As PSO shows very promising performance on SOPs with a large and complex problem landscape, many researchers start to extend PSO for tackling MOPs. The detailed review of existing MOPSOs will be provided in Section **2.3**.

In [27]–[30], the experiments have revealed that some MOPs with complex ***PF*** or ***PS*** (e.g., UF [11] and F [31] test problems) will bring significant challenges for the existing MOPSOs. To better solve these complicated MOPs, as inspired from the resource allocation strategies in MOEAs [11]–[13], a novel hybrid MOPSO is proposed with an adaptive resource allocation strategy in this paper, called HMOPSO-ARA. A novel velocity update strategy for PSO-based search and a decomposition-based clonal selection method are also presented in our algorithm to further speed up the convergence and maintain the diversity.

To comprehensively evaluate the performance of HMOPSO-ARA, this paper adopts four widely used test suits (i.e., DTLZ [32], WFG [33], UF [11] and F [31]). The experiments have justified the superiority of HMOPSO-ARA over four competitive MOPSOs (SMPSO [27], CMPSO [30], dMOPSO [29], and AgMOPSO [28]) and four state-of-the-art MOEAs (MOEA/D-ARA [34], MOEA/D-DE [31], MOEA/D-GRA [12], and EF_PD [35]).

The rest of this paper is organized as follows. Section 2 introduces some background information, i.e., the concept of PSO, decomposition approach, and some related MOPSOs in recent years. In Section 3, the details of HMOPSO-ARA are given. Section 4 shows the experimental results of HMOPSO-ARA when compared to several state-of-the-art MOEAs and MOPSOs. At last, the conclusions and future work are presented in Section 5.

II. RELATED WORK

A. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization is a popular and effective population-based optimization strategy. Let $x_i(t)$ be the position of i-th particle at t-th iteration. Then, $x_i(t)$ will be updated by using a velocity $v_i(t)$ during the search process. Then, the new position of i-th particle is calculated as follows:

$$x_i(t+1) = v_i(t+1) + x_i(t) \tag{2}$$

where $v_i(t+1)$ means the velocity for i-th particle at t-th iteration, as defined by

$$v_i(t+1) = w \cdot v_i(t) + c_1 r_1 (x_{pbest_i} - x_i(t)) + c_2 r_2 (x_{gbest_i} - x_i(t)) \tag{3}$$

where w is the inertial weight to control the impact of previous velocities, c_1, c_2 are two learning factors, r_1, r_2 are two random numbers uniformly generated within [0, 1], x_{pbest_i} and x_{gbest_i} are respectively the position information of personal best and global best particles for i-th subproblem,. In this paper, x_{gbest_i} is randomly selected from top 10% particles with the lager improvement fitness values for i-th subproblem.

B. DECOMPOSITION METHOD

The decomposition approaches [10] include the weighted sum (WS) approach, the Tchebycheff (TCH) approach and the boundary intersection (PBI) method. As the TCH method can handle the MOPs with non-convex ***PFs***, it is used in this paper as the aggregated functions, which is also widely used in most MOEAs [11]–[13], [36] and MOPSOs [28], [29], as defined by

$$\underset{x\in\Omega}{Min}\ g^{tch}(x|w, z^*) = \underset{1\leq j\leq m}{max}\{|f_j(x) - z_j^*|/w_j\} \tag{4}$$

where $z^* = \{z_1^*, \ldots, z_m^*\}$ is an approximate ideal point with $z_j^* = \min\{f_j(x)|x \in P_t\}$ for all $j = \{1, 2, \ldots, m\}$, and m is the objective number. $w = (w_1, w_2, \ldots, w_m)^T$ is the used weight vector for decomposition, having $w_j \geq 0 (j \in \{1, 2, \ldots, m\})$ and $\sum_{j=1}^{m} w_j = 1$. In case $w_j = 0$, w_j is reset to a very small value in this paper, e.g., 10^{-6}. By embedding N different weight vectors $\{w^1, w^2, \ldots, w^N\}$ in Eq. (1), a MOP can be transferred as N aggregated functions in Eq. (4). Then, the improvement value Δ_i of i-th subproblems is defined by

$$\Delta_i = \frac{g^{tch}(x_{t-1}^i|w^i, z^*) - g^{tch}(x_t^i|w^i, z^*)}{g^{tch}(x_{t-1}^i|w^i, z^*)} \tag{5}$$

where x_t^i and x_{t-1}^i are the solutions associated to the weight vectors w^i (i ={1,2, ..., m}) at t and $t-1$ generations respectively, and $g^{tch}(\cdot)$ is the TCH function as defined in Eq. (4).

C. RECENT STUDIES ON MOPSOs

In recent years, a number of MOPSOs have been presented with very promising performance [37], [38]. As shown in [39], most of MOPSOs are designed by mimicking the social cooperative and competitive behavior of bird flocking and fish schooling. Similar to MOEAs, based on the mechanisms to choose the personal-best and global-best particles, most of existing MOPSOs can also be divided into three main kinds. The first type of MOPSOs runs the selection strategies according to the non-dominance relationship, such as [27], [30], [40], the second type of MOPSOs (e.g., [29], [41]–[46]) uses the decomposition approaches to select x_{pbest_i} and x_{gbest_i}, and the third type of these MOPSOs (e.g., [47], [48]) exploits some performance indicator to run the selection procedure.

For the first type of Pareto-based MOPSOs, the dominance relationship for the particles is applied. In SMPSO [27], non-dominated particles are selected to run the PSO-based search, in which their velocities are constrained to avoid the case with too fast velocities. In CMPSO [30], non-dominated particles in an external archive are selected to run a novel co-evolution for optimizing each objective and also used to guide the PSO-based search. In AGMOPSO [40], a multi-objective gradient method was used to update the external archive, which improves the convergence speed and the local exploitation during the evolutionary procedure. Moreover, a self-adaptive flight parameters mechanism was proposed to balance the convergence and diversity according to the diversity information of the particles.

Decomposition-based MOPSOs transform MOPs into a set of subproblems using the decomposition approaches, which assign each particle to optimize one subproblem. MOPSO/D [41] may be the first try to embed the decomposition methods into MOPSOs. After that, a series of MOPSOs were published based on the decomposition framework to further improve the performance of MOPSOs. For example, in dMOPSO [29], the position of each particle was updated by a set of solutions that are selected by the decomposition method as the global best particles, while a memory re-initialization mechanism was used to enhance the diversity for particles. In MS-PSO/D [42], a generic methodology, namely MS-PSO/D, has been developed by combining PSO with MOEA/D to specifically solve combinatorial MOPs. Moreover, an element-based representation and a constructive method are utilized to generate the feasible solutions under unconstraint. Besides that, some MOPSOs were proposed by integrating the dominance relation and decomposition-based approaches. For instance, in D2MOPSO [43], MOPs will be simplified as a series of aggregation problems by using the decomposition approach, while the leader's archive is built and updated according to the dominance relation. In AgMOPSO [28], a novel decomposition approach is used in MOPSOs to select the personal-best particle and global-best particle during the evolutionary search process, while the Pareto dominance method is applied to update the external archive and particle swarms. In DP-DMOPSO [44], an external archive is adopted to reserve non-dominated solutions and a space decomposition-based mechanism is presented to renew the external archive. GSADMSPSO [45] was proposed by combining the dynamic multiple swarm particle optimization with gravitational search algorithm to enhance the ability of exploitation and exploration. In AMPSO [46], a hybrid framework using solution distribution entropy and population spacing information is designed, which can enhance the accuracy of MOPSOs and attain the final solutions with better diversity.

Indicator-based MOPSOs utilize some performance indicators (e.g., HV [19] and R2 [18]) to evaluate the performance of each particle by computing their corresponding contributions. For instance, in R2-MOPSO [47], the use of the R2 contribution in archived solutions would help to select global best leader and update swarm. A sigmoid function mapping strategy was proposed in R2HMOPSO [48], which adjusts the inertia weight and learning factors to compromise the exploration and exploitation process effectively. Moreover, in order to enhance the search ability, simulated binary crossover (SBX) operator is also utilized to reinitialize the particles. More recently, there are a number of MOPSOs further extended for tackling many-objective optimization problems (e.g., [49]–[51]).

Although the above MOPSOs have shown the promising performance on some simple MOPs, they will still encounter great challenges when solving MOPs with complex ***PF*** or ***PS***, such as UF [11] and F [31] test problems. As inspired by the above MOPSOs, this paper combines the advantages of Pareto dominance relationship and decomposition approaches to better solve the complicated MOPs. Firstly, in order to obtain an elite archive, an archive-based evolutionary search is performed on the external archive. Then an adaptive resource allocation strategy is designed in this paper to reasonably allocate the computation resource for each particle in archive according to their corresponding performance. This way, only a part of particles with good performance will be selected to run PSO-based search, which can further improve the performance on convergence and diversity. Moreover, a novel velocity update function for PSO-based search and a decomposition-based clonal selection method for archive-based search are also presented in our algorithm to further speed up the convergence and maintain the diversity.

III. THE PROPOSED HMOPSO-ARA ALGORITHM

A. THE FRAMEWORK OF PROPOSED ALGORITHM

The algorithmic framework of HMOPSO-ARA is shown in Fig. 1, which is mainly composed by the initialization, archive-based search, adaptive resource allocation process, PSO-based search and external archive update. The whole algorithm will be terminated when the termination condition is reached, and the solutions in the external archive are reported as the final result.

As illustrated in Fig. 1, firstly, the algorithm starts by initializing the population for particle swarm and external archive, and then some relevant parameters will be set simultaneously. After that, the external archive A will undergo the archive-based search to obtain a new elite population, called E. Then, the population E will combine with the external archive A to run the process of archive update. After archive-based search process, the adaptive resource allocation process is performed on the external archive A. According to the resource allocation strategy, only a part of particles in A with better performance can be selected for further optimization, i.e., PSO-based search, to approximate the true ***PF***. As the same time, each newly generated particle during PSO-based search will be added into particle swarm P. Afterward, the particle swarm P will be used to update the external archive A again. Finally, the solutions in the external archive A are outputted as the final approximate ***PF***.

In more detail, the pseudo-code of the proposed HMOPSO-ARA is clarified in **Algorithm 1**, where *gen*, *fes* and *maxfes* indicate the current generation, the current evaluation numbers and the maximum evaluation numbers, respectively. In line 1, the initialization is run, which will be

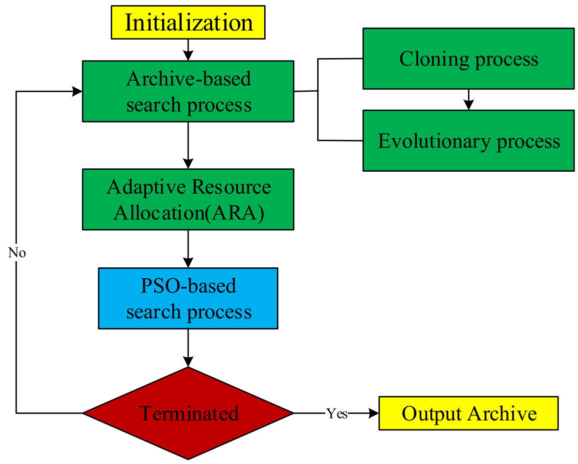


FIGURE 1. The flow chart of HMOPSO-ARA.

described in Section **3.2**. At the beginning, the archive-based search process is run as shown in line 3, which includes a decomposition-based clonal selection process and two main evolutionary operators. The details of archive-based evolutionary search are provided in Section **3.5**. Then, the newly generated population, called E, will be used to update the external archive A in line 4. After that, an adaptive resource allocation strategy will be performed on the external archive A as listed in lines 7-12, where $p(i)$ indicates the selection probability that will be assigned for each particle. Then, the computation resources will be allocated adaptively based on their corresponding selection probabilities. This adaptive resource allocation process will be introduced in Section **3.3**. This way, only part of the particles that are randomly based on their selection probabilities will be further optimized. After that, the selected particles will undergo a further evolutionary produce, i.e., PSO-based search process as listed in line 9. More details about PSO-based search will be introduced in Section **3.4**. At the same time, all the newly generated particles will be added into particle swarm, namely P. As shown in line 13, the particles P will be used to update the external archive A again, which will be introduced in Section **3.6**. Of course, the reference ideal point Z^* will be updated simultaneously as shown in lines 5 and 14. Moreover, in lines 16-20, over each of Δ_t generations (without loss of generality, we set $\Delta_t = 20$ in our algorithm), the aggregation improvement value Δ_i and the selection probability $p(i)$ will be recalculated respectively by using Eq. (5) and Eq. (6). The above evolutionary loop of HMOPSO-ARA will be terminated when the condition in line 2 is met. Finally, all the particles in the external archive A will be returned as the final approximate ***PF***.

To clarify the proposed algorithm (HMOPSO-ARA), the implementation details of other main procedures (initialization, PSO-based search, archive-based search and archive update) are respectively introduced below.

B. INITIALIZATION PROCEDURE

The pseudo-code of the initialization is described in **Algorithm 2**. First, the current generation *gen*, the current

Algorithm 1 Main Loop of HMOPSO-ARA

1: **Initialization //(Algorithm 2)**
2: **while** *fes*<*maxfes* **do**
3: use the archive A to run **Archive-based search**; /(**Algorithm 4)**
4: use the newly generated population E to **update archive** A; //(**Algorithm 5**)
5: update the ideal point Z^* simultaneously;
6: fes=fes+$|E|$;
7: **for** i= **1** to $|A|$ **then**
8: **if** $p(i) > rand$ (0,1) **then**
9: use the selected particle from A to run **PSO-based search**;// (**Algorithm 3**)
10: add the newly generated child into particle swarm P;
11: **end if**
12: **end for**
13: use the particle swarm P to **update archive** A again; //(**Algorithm 5**)
14: update the ideal point Z^* simultaneously;
15: fes=fes+$|P|$;
16: **while** $gen\%\Delta_t == 0$ **then**
17: recalculate the improvement value Δ_i for particles according to Eq.(5);
18: recalculate the selection probability $p(i)$ for particles according to Eq.(6);
19: **end while**
20: **end while**
21: **return** the final archive A;

Algorithm 2 Initialization Procedure

1: initialize the current generation, current evaluate and maximum evaluate numbers; (i.e., gen=0, fes=0, and $maxfes$ is set)
2: generate the weight vectors in objective space uniformly, i.e., $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$;
3: initialize the PSO-based population P, EA-based population E and archive A;
4: **while** each particles $i = 1$ to N **do**
5: generate a particle x_i randomly;
6: set the current velocity $v_i = 0$ and cloning number $n_i = 0$, respectively;
7: initialize the selection probability for particle x_i, i.e., $p(i) = 0$;
8: evaluate the objective value of x_i;
9: add the newly generated particle x_i into the PSO-based population P;
10: **end while**
11: initialize the neighbor sets for each particle, i.e., $B(i) = \{i_1, i_2, \ldots, i_T\}$;
12: initialize the reference ideal point for each particle, i.e., $z^* = \left\{z^1, z^2, \ldots, z^N\right\}$;
13: all the non-dominated solutions in P will be added into the external archive A;

evaluate numbers *fes* and the maximum evaluate numbers *maxfes* are initialized in line 1. As shown in line 2, a set of N weight vectors $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$ is uniformly generated to decompose a MOP into a set of SOPs by using Eq. (4). Second, since no prior knowledge of the search landscape is available, an initial swarm $P = \{x_1, x_2, \ldots, x_N\}$ is randomly sampled in decision space Ω in lines 4-10, while v_i, n_i and $p(i)$ respectively indicate the current velocity, the cloning number and the selection probability based on its corresponding fitness value for each particle, which are initialized in lines 6-8. Then, as shown in line 11, the T closest weight vectors $w^{i1}, w^{i2}, \ldots, w^{iT}$ according to the Euclidean distances between each weight vector w_i and other weight vectors will be selected to compose the neighbor set $B(i)$. Moreover, as the true ideal point cannot be obtained at the beginning, an approximated point is used instead, which will find the minimum value of each objective in line 12, i.e., $z_i^* = \min\{f_i(x)|x \in P_t\}$ for all $i = \{1,2, \ldots, m\}$. At last, all the non-dominated particles in P will be added into the external archive A in line 13.

C. AN ADAPTIVE RESOURCE ALLOCATION STRATEGY

Generally speaking, during the evolutionary process, all the individuals will be equally treated, which will obtain the same amount of computation resources. Whereas, it is normal that the different parts of ***PF*** for one MOP have different difficulties to approximate, hence different individuals usually have different contributions on optimization.

As inspired by the phenomenon in many MOPs, we consider organizing the computation resources more reasonably according to the different performance of each subproblem. Hence, we attempt to employ the dynamic resource allocation approach into traditional MOPSO algorithm, which is similar to these MOEAs based on resource allocation, such as MOEA/D-DRA [12], MOEA/D-GRA [11] and MOEA/D-IRA [13]. In our proposed algorithm, according to the relative improvement values, a selection probability $p(i)$ is designed for each particle to judge whether each particle should be selected or not, calculated as below.

$$p(i) = (1 - p_{min}) \times \frac{1}{1.0 + p_{min} \times \exp(20 \times \frac{r_i}{T} - 0.5)} \quad (6)$$

where T and r_i respectively indicate the neighbor size and the local rank based on the relative improvement values Δ_i among the corresponding neighboring set $B(i)$. p_{min} means the minimum selection probability for each particle, which is used to ensure that each particle will have an opportunity to be selected (in our algorithm, p_{min} is set as 0.15). For a visual observation, we plot the dynamic change tendency between the selection probability $p(i)$ and local rank r_i, as shown in Fig. 2.

It is easy to learn from Fig. 2 that the selection probability and the corresponding local rank for each particle

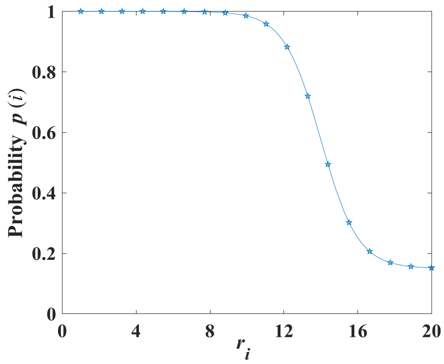


FIGURE 2. **The dynamic change tendency between the selection probability $p(i)$ and the corresponding local rank r_i.**

are positively correlated. That is to say, these particles with higher local rank, which are ranked by the relative improve values Δ_i, will also have bigger selection probabilities than others with a lower local rank. This way, these particles will have a higher selection probability to be selected in the next generation for further optimization, as they have larger Δ_i values. On the contrary, these particles are seldom selected in the next generation if their relative improve values Δ_i are low. Thus, regarding all the particles, they will be allocated with different computation resources according to their different relative improvement values. In other word, more evolutionary computation resources will be allocated to optimize these particles with the larger relative improvement on aggregated function values.

D. PSO-BASED SEARCH PROCEDURE

In this part, the PSO-based search procedure is performed. As introduced in most of existing traditional MOPSOs, the positional information of personal-best and global-best particles, namely x_{pbest_i} and x_{gbest_i}, are used to update the velocities of the particles. Whereas, it may encounter some difficulties on tackling some MOPs with complicated ***PFs*** or ***PSs*** for most existing traditional velocity update strategies. In order to enhance the robustness and performance of MOPSOs, we propose a novel velocity update method in this paper, which not only can provide another new search direction from x_{pbest_i} to x_{gbest_i} to speed up the convergence, but also can make more disturbances on the particles to improve the diversity, as defined below.

$$v_i(t+1) = w \cdot v_i(t) + c_1 r_1 (x_{pbest_i} - x_i(t)) + c_2 r_2 (x_{lbest_i} - x_{gbest_i}) + c_3 r_3 (x_{gbest_i} - x_{pbest_i}) \tag{7}$$

where t is the current iteration number; w is the inertial weight; c_1, c_2 and c_3 are three controlling factors; r_1, r_2 and r_3 are three uniformly distributed random numbers from 0 to 1. x_{pbest_i} and x_{gbest_i} respectively indicate the position information of personal-best particle and global-best particle, which have been shown in Eq. (4). Besides that, in order to enhance the search ability for PSO-based search, x_{lbest_i} is

Algorithm 3 PSO-Based Search Procedure

1: select p_{best_i}, g_{best_i} and $lbest_i$ from the external archive A respectively;
2: update the particle's velocity v_i using Eq.(7);
3: update the particle's position x_i using Eq.(2);
4: recalculate the particle's objective values;

employed in our proposed velocity update function, which is selected from its neighborhood as the position information of local best particle. This way, the particle's position will be updated according to x_{pbest_i}, x_{gbest_i} and x_{lbest_i} simultaneously, as defined in Eq. (2).

To have a comprehensive understanding of our proposed velocity update strategy, its design details are introduced below. Totally, our proposed novel particle velocity strategy is composed by four main parts.

Regarding the first part $w \cdot v_i(t)$, it indicates the "inertial" part as introduced in Eq.(3) of Section **2.1**. For the second part $c_1 r_1 (x_{pbest_i} - x_i(t))$, it provides a search direction from the current particle to the personal best one for i-th subproblem. Regarding to the third part $c_2 r_2 (x_{lbest_i} - x_{gbest_i})$, the main purpose of the design is to enhance the disturbance ability of the PSO-based search strategy, which is similar to the features of differential evolution (DE) operator as illustrated in [4]. The last part $c_3 r_3 (x_{gbest_i} - x_{pbest_i})$ can provide one novel search direction between the personal-best x_{pbest_i} and global-best x_{gbest_i}. The newly embedded search direction from x_{pbest_i} to x_{gbest_i} can push the search towards the global best particles, which is favorable for convergence.

This way, the proposed novel velocity update strategy not only has a strong ability to converge quickly, but also has a strong disturbance ability to search. It should be pointed out that the disturbance ability is very important for the PSO-based search as a strong disturbance ability is suitable for solving these MOPs with complicated ***PFs*** or ***PSs***.

The pseudo-code of PSO-based search is illustrated in **Algorithm 3**. At the beginning, in its line 1, the selection procedures are run, in which $pbest_i$, $gbest_i$ and $lbest_i$ are respectively picked up from the external archive A. After that, as shown in lines 2-3, the information of velocity and position are recalculated by using Eq. (2) and Eq. (7), respectively. Then, the objective values of all the particles in particles P are reevaluated as listed in line 4 simultaneously. Terminally, the newly generated particles P will be returned.

E. ARCHIVE-BASED SEARCH PRODUCE

In this subsection, archive-based search process is performed on the external archive A, which is composed by two main produces. They are a decomposition-based clonal selection strategy and two evolutionary-based operators (i.e., simulated binary crossover (SBX) and polynomial-based mutation (PM) [52]), respectively. The details of archive-based search process are introduced below.

1) DECOMPOSITION BASED CLONAL SELECTION

In most of existing MOIAs [53]–[55], the clonal selection strategy has been validated to speed up the convergence. More recently, a novel clonal selection approach, namely DCSS, based on the framework of decomposition is first proposed in MOIA-DCSS [56], which has been confirmed that it can enhance the ability of algorithms for solving some complicated MOPs.

Hence, the decomposition-based clonal selection method is also implemented in this paper to optimize the archive A. As illustrated in MOIA-DCSS [56], all the cloning number for each solution is calculated by the relative improvement values instead of the crowding distance, which can enhance the convergence pressure of our algorithm. It plays a crucial role in tacking some MOPs with variable linkages (i.e., UF1-UF10 and F1-F9 test problems).

Some details of decomposition-based clonal selection strategy (DCSS) are briefly introduced below. Assume that C and N are the cloning population and its size, respectively. The elitist population with the biggest relative improvement values selected from external archive A for cloning is PC and n_A indicates the corresponding size of the elitist population with a higher improvement values on aggregated function. It should be noticed that n_A is smaller than N (we set as $N/5$ in this paper). The mathematical model of proportional cloning as defined by

$$C = \bigcup_{i=1}^{n_A} \{n_i \otimes a_i\} \tag{8}$$

where operation $n_i \otimes a_i$ indicates to copy the individual, and n_i is the number of clones for each individual in the external archive A, which is computed by

$$n_i = \left\lceil N \times \frac{\Delta_i}{\sum_{j=0}^{N} \Delta_j} \right\rceil \tag{9}$$

where N is the cloning population size and Δ_i is the relative improvement values as defined in Eq. (5). This way, these solutions with larger improvement values will have more clones in the next generation, which can speed up the convergence of the entire population.

2) THE PROCESS OF EVOLUTIONARY SEARCH

After cloning, the newly generated population C will undergo two evolutionary operators, i.e., SBX and PM, a brief introduction of which is given below. Let us assume that the individuals in population C is $x^i = (x^1, x^2, \ldots, x^n)$ and other parents $x^{r_i} = (x^{r_1}, x^{r_2}, \ldots, x^{r_3})$ are randomly selected from the external archive A.

1) Simulated binary crossover:

$$\begin{aligned} z_i^0 &= 0.5 \times [(w_i + v_i) - \beta_0 \times (w_i - v_i)] \\ z_i^1 &= 0.5 \times [(w_i + v_i) - \beta_1 \times (w_i - v_i)] \end{aligned} \tag{10}$$

where z_i^0 and z_i^1 are two decision parameters of the generated offspring, w_i and v_i indicate the maximum and minimum values of x^i and x^{r_i}, respectively. $\beta_i (j = 0, 1)$ are computed as follows:

$$\beta_i = \begin{cases} [r_j \times a_j]^{1/(\eta+1)}, & \text{if } r_i \le 1/a_j \\ \left[\dfrac{1}{2 - r_j \times a_j}\right]^{1/(\eta+1)}, & \text{otherwise} \end{cases} \tag{11}$$

where $r_j (j = 0, 1)$ are evenly distributed random numbers in $[0, 1]$, η is a crossover distribution index, and $a_j (j = 0, 1)$ is defined below

$$a_j = \begin{cases} \left(2 - \left(1 + 2 \times \dfrac{v_i - l_i}{w_i - v_i}\right)\right)^{-(\eta+1)}, & j = 0 \\ \left(2 - \left(1 + 2 \times \dfrac{u_i - w_i}{w_i - v_i}\right)\right)^{-(\eta+1)}, & j = 1 \end{cases} \tag{12}$$

where l_i and u_i are the lower and upper bounds for the i-th decision variable, respectively.

Polynomial-based

2) mutation:

After the above SBX operator, the newly generated individuals, defined as $v^i = (v^1, v^2, \ldots, v^n)$, are further permuted to get offspring solutions y^i $(y^i = y^1, y^2, \ldots, y^n)$ by using PM operator, defined as

$$y_j^i = \begin{cases} v_j^i + \delta_j \times (u_i - l_j) & \text{if } rand(0, 1) < p_m \\ v_j^i & \text{otherwise} \end{cases} \tag{13}$$

where p_m indicates the mutation probability, and δ_j is calculated by

$$\delta_j = \begin{cases} \left[2r + (1 - 2r) \times (\dfrac{u_j - u_i}{u_j - l_i}))^{1/(\eta+1)}\right], \\ \quad \text{if } rand(0, 1) < 0.5 \\ 1 - \left[2r + (1 - 2r) \times (\dfrac{u_j - u_i}{u_j - l_i}))^{(\eta+1)}\right]^{1/(\eta+1)}, \\ \quad \text{otherwise} \end{cases} \tag{14}$$

where r is a evenly distributed random number in $[0, 1]$ and η is a mutation distribution index.

3) THE COMPLETE PROCESS OF ARCHIVE-BASED SEARCH

The pseudo-code for the external archive-based search is described in **Algorithm 4**. In lines 1-4 about the clonal selection, the cloning number n_i for each subproblem in the external archive A is calculated using Eq. (9) according to their corresponding relative improvement value Δ_i defined in Eq. (5). Then, the cloning produce is performed to duplicate the solutions according to Eq. (8) based on the cloning number n_i. Thus, more offspring will be generated from these solutions with a higher improvement value. In other word, they are easier to be promoted and further optimized in the next generation. After the clonal selection procedure, the newly generated cloned population C will run two traditional evolutionary operators (i.e., SBX and PM). As shown in lines 5-11 about the evolutionary search. First, one solution x^{r_1} is randomly selected from the cloning population C as a parent solution. Then, two solutions (x^{r_1} and x^i) will undergo

Algorithm 4 Archive-Based Search Procedure

```
while i = 1 to |A| do
    the cloning number n_i for each solution is
    computed according to Δ_i using Eq.(9);
end while
use Eq.(8) on A to generate the cloned population C;
while i = 1 to|C| do
    one solutionx^{r_1} is randomly picked up from the
    cloned population C;
    x^i and x^{r_1} run SBX operators to generate a child,
    namely v^i;
    polynomial mutation (PM) is performed on v^i to
    generate a new solution y^i;
    the objective values of y^i are recalculated;
    the newly generated child y^i is added into
    population E;
end while
returnE
```

Algorithm 5 Archive Update Procedure

```
1: while i = 1 to |E| or |P| do
2:     add all the non-dominance solutions in E or P
       into external archive A;
3: end while
4: if |A| > N then
5:     calculate the improvement value for each
       solution using Eq.(3),
       eliminate the biggest one in the external
       archive A;
6: end if
7: return external archive A;
```

SBX and PM, which can make more disturbances on the external archive A. Finally, the evolutionary population E composed with all the newly generated solutions will be returned. Thus, the convergence and diversity of external archive are enhanced by employing the archive-based evolutionary search processes.

F. ARCHIVE UPDATE

After performing the PSO-based search and archive-based search processes, a new particle swarm (P) and population (E) are produced, respectively. In order to control a certain number of elitist solutions in the external archive (A), it is necessary to employ an approximate selection mechanism to update the external archive. As a result, the search direction can be effectively guided to approximate the true ***PF***. The pseudo-code of the archive update is described in **Algorithm 5** below.

Firstly, all the non-dominance solutions in the population E or particle swarm P will be added into external archive A in line 2. Then, the archive update procedure will be performed to ensure the fix size of archive if the size of external archive $|A|$ is more than the maximum size N that is predefined. As shown in lines 4-6, the individual in the external archive with the biggest relative improvement value will be deleted until the size of archive equals to the predefined size. Finally, the newly updated external archive A will be outputted.

IV. EXPERIMENTAL RESULTS

A. TEST PROBLEMS

In our experimental comparison, there are thirty-five different test MOPs employed to evaluate the performance and superiority of HMOPSO-ARA, and their features for each class of test problems are briefly introduced below.

UF1-UF10 have very complicated ***PSs*** in the search space, which are initially proposed as the benchmark problems for the competition of MOEAs in CEC 2009 [11]. Similar to the UF test problems, the F1-F9 test problems are first proposed in [31], which aim to estimate the ability of MOEAs in tackling complicated ***PS*** shapes. Moreover, the common feature between the UF and F test suits is that they have strong variable linkages in decision space. For DTLZ1-DTLZ9 in [32], different problems have different difficulties for approximating the true ***PFs***. Considering the WFG test problems (WFG1-WFG9) [33], they are all characterized with various complex features and mixed ***PF*** shapes.

Moreover, from the perspective of the objective numbers, they can be divided into two categories, two objectives (e.g., WFG1-WFG9, UF1-UF7, F1-F5 and F7-F9 test problems) and three objectives (e.g., DTLZ1-DTLZ9, UF8-UF10 and F6 test problems). All the test problems used in this paper have various complex characteristics and complicated ***PSs***. The numbers of decision variables are set to 30 for DTLZ1-DTLZ9, UF8-UF10 and F6 test problems, and set to 10 for UF1-UF7, F1-F5 and F7-F9 test problems. It is worth noticing that the numbers of position-related and distance-related decision variables in WFG test problems are respectively set to 4 and 20.

B. PERFORMANCE INDICATORS

As no single performance indicator is able to make a comprehensive measure on the performance of an algorithm, two commonly used performance indicators (i.e., inverted generational distance (IGD) [57] and hypervolume (HV) [19]) are employed to measure the performance of all the compared algorithms.

1) Inverted generational distance (IGD): The IGD indicator calculates the average distance from a set of uniformly distributed solutions along the true ***PF*** to the approximation set that are obtained by the compared algorithms. The IGD value from S to S', namely $IGD(S, S')$, can be calculated as

$$IGD(S, S') = \frac{\sum_{i=1}^{|S|} dist(S_i, S')}{|S|} \tag{15}$$

where $dist(S_i, S')$ is the minimal Euclidean distance between S_i and any point in S', and $|S|$ indicates the cardinality of S. The true ***PF*** of the underlying MOP has to be known in advance when computing the IGD metric. A smaller $IGD(S, S')$ value is considered to be a better performance of S' to approximate S.

TABLE 1. The parameter settings of all the compared algorithms.

Algorithms	Parameter settings
SMPSO	$N=100, \omega\in[0.1,0.5], c_1,c_2\in[1.5,2.5], P_m=1/\eta, \eta_m=20$
AgMOPSO	$N=100, P_c=0.9, P_m=1/n, \eta_c=20, \eta_m=20, \omega\in[0.1,0.5], F_2=0.5, T=20$
CMPSO	$N=100, \omega\in[0.1,0.5], c_1,c_2\in[1.5,2.5]$
dMOPSO	$N=100, \omega\in[0.1,0.5], c_1,c_2\in[1.5,2.5]$
MOEA/D-GRA	$N=100, P_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$
MOEA/D-DE	$N=100, P_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2, CR=1.0, F=0.5$
MOEA/D-ARA	$N=100, P_m=1/n, \eta_m=20, T=20, CR=1.0, F=0.5, \delta=0.9, n_r=2$
EF_PD	$N=100, P_c=0.9, P_m=1/n, \eta_c=20, \eta_m=20, T=20, n_r=2, \delta=0.9, F=0.5, CR=1$
HMOPSO-ARA	$N=100, P_c=0.9, P_m=1/n, \eta_c=20, \eta_m=20, \omega\in[0.1,0.5], c_1,c_2,c_3\in[1.5,2.5], T=20$

2) Hyper-volume (HV): The HV metric measures the size of the objective space dominated by the solutions in S and bounded by the reference point $z^r = (z_1^r, z_2^r, \ldots, z_m^r)^T$, as defined by

$$HV(S) = VoI\left(\bigcup_{x\in S}\left[f_1(x), z_1^r\right] \times \ldots \times \left[f_m(x), z_m^r\right]\right) \quad (16)$$

where $VoI(.)$ indicates the Lebesgue measures. These points that cannot dominate the reference point will be deleted when considering the HV metric. A larger HV value implies a better performance.

In our experiments, In order to calculate the HV values, the relative reference points are set as follows. For DTLZ1-DTLZ7 test problems, (1.0, 1.0, 1.0) is set for DTLZ1 and (2.0, 2.0, 2.0) is set for DTLZ2-DTLZ6. For all the WFG test problems with two objectives, we set (3.0, 5.0) as the reference point. For other problems, (2.0, 2.0) is set for these test suits with two objectives (i.e., UF1-UF7, F1-F5 and F7-F9 test problems) and (2.0, 2.0, 2.0) is set for three-objective test problems (i.e., UF8-UF10 and F6 test problems).

C. EXPERIMENTAL SETTING

In this paper, HMOPSO-ARA is compared with several state-of-the-art MOEAs (i.e., MOEA/D-ARA [34], MOEA/D-DE [31], MOEA/D-GRA [12] and EF_PD [35]), and four competitive MOPSOs (i.e., SMPSO [27], dMOPSO [29], CMPSO [30] and AgMOPSO [28]). All the compared algorithms are validated to show the superior performance in solving numerous MOPs (i.e., DTLZ1-DTLZ7, WFG1-WFG9, UF1-UF10 and F1-F9 test problems). Obviously, the comparisons of HMOPSO-ARA with these algorithms as mentioned above are very comprehensive and convincing. To be fair, all the relative parameters of all the algorithms are set according to the introduction in the corresponding references, which are summarized in Table 1.

As shown in Table 1, N is the population size; ηc and ηm respectively indicate the distribution indexes of SBX and PM; p_c and p_m are the probabilities to run crossover and mutation, respectively. T is the size of neighborhood regarding the weight vectors, while δ and n_r respectively indicate the probability to select parent solutions from T neighbors and the maximum number of parent solutions that are replaced by each child solution. Moreover, ω, $c1$, $c2$ and $c3$ are the parameters in the velocity update equation for MOPSOs.

Please note that the settings of $N = 100$ listed in Table 1 are only for the WFG test problems with the maximum number of function evaluations as 25 000. When solving other test problems, the population size (N) and the maximum number of function evaluations (*maxfes*) are set according to their different features (e.g., the difficulty and the complexity of test problems). For F1-F5, F7-F9 and UF1-UF7 as bi-objective test problems, their population sizes $N = 300$ and *maxfes*= 150 000. For F6 and UF8-UF10 as three-objective test problems, their population sizes $N = 600$ and *maxfes*= 300 000. When considering all the DTLZ test problems, their population sizes $N = 105$ and *maxfes*= 52 500, respectively.

All the experiments were independently run 30 times with different random seeds. The mean values and the standards deviations (included in brackets after the mean results) of IGD and HV in 30 runs were collected for comparison. In order to have an obvious observation on the best performance, the boldface and gray background in all the experimental comparison tables indicates the best mean value for each problem. Furthermore, to obtain a statistically sound conclusion, Wilcoxon rank sum test was run with a significance level $\alpha = 0.05$ to show the statistically significant differences between the results of HMOPSO-ARA and other competitors. In the following tables, the symbols "+", "−", and "∼" indicate that the results of other competitors are significantly better than, worse than, and similar to the ones of HMOPSO-ARA using this statistical test, respectively.

D. COMPARISON OF HMOPSO-ARA AND FOUR PEER MOEAs

In this part, four state-of-the-art MOEAs, i.e., MOEA/D-ARA [34], MOEA/D-DE [31], MOEA/D-GRA [12] and EF_PD [35], are used to compare with HMOPSO-ARA. It should be pointed that the IGD results listed in Table 2 are obtained after 30 independent runs.

It is obvious that the proposed HMOPSO-ARA shows superiority over other peer algorithms based on the mean IGD values. As shown in Table 2, HMOPSO-ARA can

TABLE 2. Experimental results (mean and standard deviation) of IGD values on all test problems obtained by MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA, EF_PD and HMOPSO-ARA.

Instances	MOEA/D-ARA	MOEA/D-DE	MOEA/D-GRA	EF-PD	HMOPSO-ARA
DTLZ1	3.431E-01(3.093E-01)-	1.909E-01(1.741E-03)-	3.018E-02(7.406E-04)-	2.228E-02(9.529E-04)-	**1.990E-02(6.641E-04)**
DTLZ2	5.259E-02(4.472E-04)~	6.747E-02(3.851E-04)-	6.756E-02(3.142E-04)-	5.999E-02(2.110E-03)-	**5.236E-02(7.782E-04)**
DTLZ3	1.857E+00(3.984E+00)-	1.110E-01(2.636E-03)-	7.079E-02(1.493E-03)-	6.119E-02(2.229E-03)-	**5.290E-02(7.662E-04)**
DTLZ4	**3.407E-02(7.533E-03)+**	5.800E-02(1.142E-02)-	5.678E-02(1.085E-02)-	6.121E-02(4.961E-03)-	3.895E-02(2.739E-03)
DTLZ5	4.495E-02(1.298E-03)-	1.256E-02(4.157E-05)-	1.256E-02(3.221E-05)-	**3.861E-03(9.624E-05)+**	3.961E-03(1.592E-04)
DTLZ6	4.377E-02(1.552E-04)-	1.192E-02(2.094E-05)-	1.192E-02(1.801E-05)-	**3.643E-03(1.499E-04)~**	3.737E-03(2.186E-04)
DTLZ7	1.273E-01(6.962E-03)-	2.227E-01(1.230E-02)-	1.971E-01(2.323E-03)-	1.483E-01(1.058E-02)-	**5.826E-02(1.901E-03)**
WFG1	1.167E+00(1.060E-02)-	1.153E+00(1.107E-01)-	1.089E+00(1.598E-01)-	7.102E-02(5.903E-02)-	**1.235E-02(2.634E-04)**
WFG2	7.687E-02(5.031E-02)-	7.087E-02(4.827E-02)-	6.153E-02(4.477E-02)-	5.182E-02(5.473E-02)-	**1.019E-02(2.805E-04)**
WFG3	1.605E-02(1.299E-03)-	1.485E-02(6.985E-04)-	1.481E-02(7.021E-04)-	1.212E-02(3.974E-04)-	**1.179E-02(4.199E-04)**
WFG4	6.221E-02(7.185E-03)-	5.289E-02(9.372E-03)-	4.930E-02(7.551E-03)-	1.125E-02(5.782E-04)-	**1.075E-02(2.992E-04)**
WFG5	6.716E-02(9.167E-05)-	6.717E-02(1.600E-04)-	6.717E-02(2.231E-04)-	6.635E-02(6.421E-05)-	**6.585E-02(8.294E-05)**
WFG6	**2.523E-02(1.332E-02)+**	3.639E-02(1.111E-02)+	6.213E-02(2.211E-02)-	3.879E-02(3.238E-02)~	5.039E-02(4.134E-02)
WFG7	1.754E-02(7.154E-04)-	1.732E-02(3.427E-04)-	1.690E-02(2.699E-04)-	1.222E-02(1.505E-04)-	**1.206E-02(2.231E-04)**
WFG8	2.313E-01(1.562E-02)-	2.187E-01(3.614E-02)~	**1.873E-01(6.407E-02)~**	2.166E-01(5.126E-03)~	2.048E-01(2.021E-02)
WFG9	1.025E-01(2.602E-01)-	5.004E-02(1.354E-03)-	9.019E-02(1.361E-01)-	7.458E-02(2.565E-02)-	**4.466E-02(5.105E-03)**
UF1	2.973E-03(2.146E-04)+	1.741E-03(7.007E-05)+	**1.711E-03(1.015E-04)+**	5.728E-03(5.772E-05)-	5.679E-03(4.385E-04)
UF2	1.291E-02(3.464E-03)-	7.153E-03(1.873E-03)-	**3.916E-03(1.219E-03)+**	7.782E-03(2.588E-03)-	5.008E-03(4.866E-04)
UF3	3.295E-02(3.847E-02)-	1.314E-02(1.583E-02)~	**3.547E-03(2.672E-03)+**	1.516E-02(1.866E-02)-	7.594E-03(4.383E-03)
UF4	6.409E-02(6.642E-03)-	6.068E-02(5.068E-03)-	5.765E-02(4.692E-03)-	**3.630E-02(4.235E-04)+**	3.894E-02(6.673E-04)
UF5	3.747E-01(8.295E-02)-	2.957E-01(8.175E-02)~	3.027E-01(8.719E-02)~	**2.509E-01(1.892E-01)~**	2.666E-01(2.353E-01)
UF6	1.690E-01(8.629E-02)~	1.679E-01(1.227E-01)~	**1.828E-01(1.025E-01)+**	2.249E-01(2.335E-01)~	2.838E-01(3.349E-01)
UF7	3.417E-03(6.365E-04)-	**2.348E-03(3.830E-04)+**	2.563E-03(2.011E-04)+	2.643E-03(3.580E-04)+	2.905E-03(2.045E-04)
UF8	**3.964E-02(5.328E-03)+**	6.465E-02(2.010E-02)~	6.881E-02(2.310E-02)-	8.249E-02(6.454E-03)-	5.685E-02(3.116E-03)
UF9	8.090E-02(1.033E-01)~	7.355E-02(1.021E-01)~	8.164E-02(1.065E-01)-	8.448E-02(1.129E-01)~	**7.007E-02(7.209E-02)**
UF10	4.659E-01(6.828E-02)-	5.043E-01(6.523E-02)-	4.284E-01(6.177E-02)-	3.001E-01(1.105E-01)-	**2.411E-01(2.211E-02)**
F1	1.341E-03(3.118E-05)-	1.304E-03(4.152E-05)-	1.284E-03(3.221E-05)~	**1.225E-03(2.503E-05)+**	1.272E-03(2.417E-05)
F2	7.650E-02(1.089E-01)-	4.938E-02(7.385E-02)-	**6.393E-03(7.762E-04)+**	2.237E-02(2.052E-03)-	8.668E-03(3.165E-04)
F3	8.316E-02(1.488E-01)-	2.154E-02(2.068E-02)-	**2.090E-03(3.902E-04)+**	4.079E-03(2.806E-04)+	4.946E-03(3.484E-04)
F4	3.299E-02(5.129E-03)-	7.577E-03(1.665E-03)-	**1.771E-03(1.393E-04)+**	1.877E-03(1.252E-04)+	5.879E-03(6.094E-04)
F5	4.672E-02(6.865E-02)-	1.546E-02(4.522E-03)-	**4.180E-03(1.050E-03)+**	7.118E-03(1.529E-03)~	6.640E-03(3.709E-04)
F6	**2.215E-02(2.445E-04)+**	3.060E-02(1.305E-03)-	3.044E-02(2.115E-03)-	3.508E-02(4.203E-03)-	2.955E-02(1.022E-03)
F7	1.353E-02(1.459E-02)-	9.146E-03(9.370E-03)~	1.572E-03(1.816E-04)+	**1.444E-03(1.152E-04)+**	3.645E-03(9.133E-04)
F8	1.603E-01(5.580E-02)-	8.594E-02(3.274E-02)-	**8.843E-03(2.807E-03)+**	2.864E-02(1.609E-02)~	4.099E-02(1.947E-02)
F9	3.294E-02(1.756E-02)~	3.505E-02(1.180E-02)~	**2.167E-03(3.707E-04)+**	3.422E-02(2.927E-04)-	1.401E-02(3.816E-03)
Best/All	4/35	1/35	11/35	6/35	13/35
Total	26-/4~/5+	24-/8~/3+	20-/3~/12+	20-/8~/7+	/

+: HMOPSO-ARA shows significantly better performance in the experimental comparison.

-: . HMOPSO-ARA shows significantly worse performance in the experimental comparison.

~: There is no significant difference between the compared experimental results.

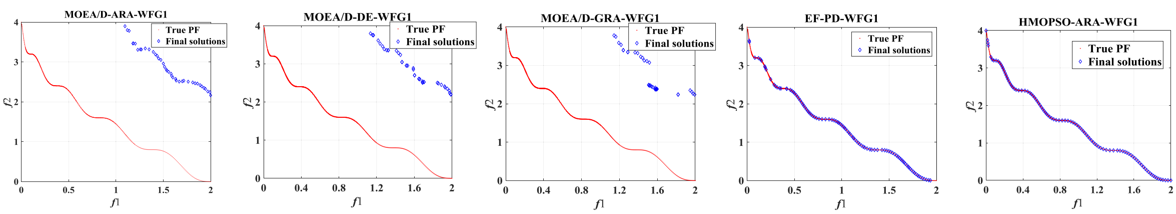


FIGURE 3. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on WFG 1.

obtain best results on 13 cases among all the 35 cases. While, MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD can only perform best on 4, 1, 11 and 6 test instances, respectively. When considering the WFG test problems, HMOPSO-ARA also shows the obvious advantages, as HMOPSO-ARA obtains all the best results except WFG6 and WFG8, while MOEA/D-ARA and MOEA/D-GRA perform best respectively on WFG6 and WFG8. Moreover, when considering the DTLZ test problems, our proposed algorithm HMOPSO-ARA can also show

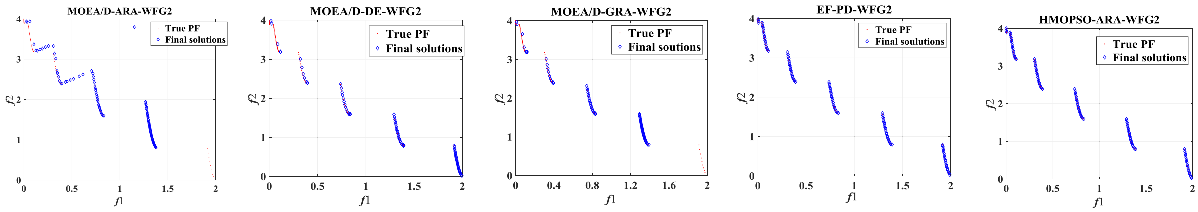


FIGURE 4. **Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on WFG2.**

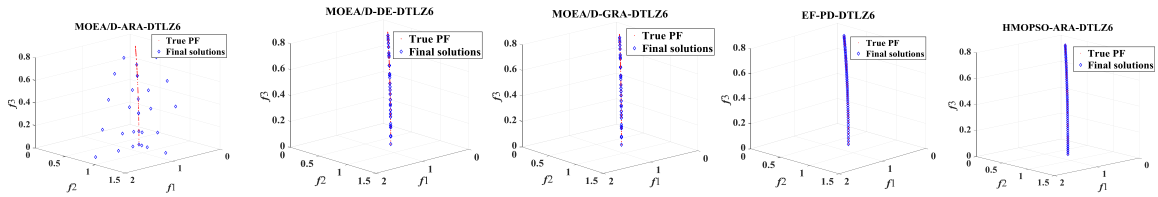


FIGURE 5. **Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on DTLZ6.**

obvious superiority over the peer algorithms, as HMOPSO-ARA can obtain best performance on 4 cases among all the 7 DTLZ test problems and EF_PD can only obtain the best results on DTLZ4-DTLZ5, while MOEA/D-ARA performs best on DTLZ4. However, when tackling the UF and F test problems with complicated ***PFs***. HMOPSO-ARA is statistically similar to the peer algorithms according to the IGD results, as all the peer algorithms are proposed under the decomposition framework and use the differential evolution operator during their evolutionary process. As shown in Table 2, HMOPSO-ARA, MOEA/D-ARA MOEA/D-DE, MOEA/D-GRA and EF_PD can respectively obtain the best results on 2, 2, 1, 10, and 4 cases among all the ten UF test problems and nine F test problems. This superiority of MOEA/D-GRA on solving UF and F test problems with variable linkages is mainly due to the employment of the adaptive resource allocation strategy, which has been validated to have some advantages in solving some complicated MOPs, where different ***PF*** parts have different difficulties to be converged. At last, in the last row of Table 2, the experimental results of HMOPSO-ARA and peer algorithms are summarized, which shows the statistical quantity of the problems that HMOPSO-ARA outperforms (-), is similarly to ($\sim$) and underperforms (+) the compared algorithms. In summary, as we can learn from the results of Wilcoxon's rank sum test, HMOPSO-ARA shows superiority over MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD on 26, 24, 20 and 20 cases among 35 test problems, respectively. On the contrary, HMOPSO-ARA is only worse than MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD respectively on 5, 3, 12 and 7 test problems.

Therefore, a conclusion can be drawn from Table 2 that the proposed HMOSPO-ARA shows obvious superiority over other compared algorithms when considering all the test problems adopted in Table 2, as HMOPSO-ARA obtained the best results on most test problems regarding IGD.

To visually show and support the above discussions, Figs. 3-5 are listed below, which give the plots of the final approximated non-dominated solutions obtained by MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD and HMOPSO-ARA on three test MOPs (i.e., WFG1, WFG2 and DTLZ6).

We can draw conclusions from Figs. 3-5 that HMOPSO-ARA obviously outperforms MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD, as the populations obtained by HMOPSO-ARA are more smooth and closer to the true ***PFs*** when compared to that of other compared algorithms. As shown in Fig. 2 for WFG1, it is easy to conclude that the convergence pressure of HMOPSO-ARA is stronger than that of other compared algorithms, as their final solution sets are far from the true ***PF***. On the contrary, the final solution set obtained by HMOPSO-ARA is much closer to the true ***PF***. Moreover, in Figs. 4-5 for WFG2 and DTLZ6 respectively, the final solution sets obtained by our proposed HMOPSO-ARA can be distributed more smoothly and completely on the entire true ***PFs***, while MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD show some difficulties in finding the complete ***PFs***, as their final solutions fail to cover the whole true ***PFs***.

The HV results of HMOPSO-ARA and the peer algorithms on all the test problems (i.e., DTLZ1-DTLZ7, WFG1-WFG9, UF1-UF10 and F1-F9 test instances) are summarized in Table 3. Some conclusions also can be learned from the experimental comparison results. In summary, HMOPSO-ARA can perform best on 17 cases among all the 35 cases according to the values of HV. Whereas, MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD can only respectively show the best performance on 2, 1, 9 and 6 test problems out of all the 35 test problems. Therefore, HMOPSO-ARA shows the obvious superiority when compared with the mentioned competitors. Moreover, according to the summary in the last row of Table 3,

TABLE 3. Experimental results (mean and standard deviation) of HV values on all test problems obtained by MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA, EF_PD and HMOPSO-ARA.

Instances	MOEA/D-ARA	MOEA/D-DE	MOEA/D-GRA	EF-PD	HMOPSO-ARA
DTLZ1	7.357E-01(3.182E-01)-	8.780E-01(1.014E-02)-	9.671E-01(1.215E-03)-	9.703E-01(7.064E-04)-	**9.735E-01(2.426E-04)**
DTLZ2	7.399E+00(3.789E-03)-	7.377E+00(2.266E-03)-	7.377E+00(2.236E-03)-	7.390E+00(4.172E-03)-	**7.414E+00(1.062E-03)**
DTLZ3	3.937E+00(7.385E+00)-	7.366E+00(1.115E-02)-	7.369E+00(8.839E-03)-	7.388E+00(4.257E-03)-	**7.413E+00(2.697E-03)**
DTLZ4	7.403E+00(2.812E-03)-	7.381E+00(3.892E-03)-	7.370E+00(7.349E-03)-	7.392E+00(5.875E-03)-	**7.416E+00(1.471E-03)**
DTLZ5	2.189E+01(1.267E-01)-	2.209E+01(1.018E-04)-	2.209E+01(2.300E-04)-	**2.210E+01(1.040E-04)+**	2.210E+01(3.921E-04)
DTLZ6	2.064E+01(5.185E-01)-	2.209E+01(1.846E-04)-	2.209E+01(1.783E-04)-	**2.210E+01(6.894E-05)+**	2.210E+01(4.409E-05)
DTLZ7	9.391E+00(1.376E-02)-	9.271E+00(9.293E-03)-	9.271E+00(5.361E-03)-	8.724E+00(1.293E-01)-	**9.478E+00(1.291E-02)**
WFG1	5.565E+00(6.341E-02)-	5.698E+00(4.906E-01)-	5.951E+00(8.171E-01)-	1.173E+01(4.682E-01)-	**1.207E+01(3.312E-04)**
WFG2	1.039E+01(8.874E-01)-	1.099E+01(8.350E-01)-	1.103E+01(8.029E-01)-	1.088E+01(8.292E-01)-	**1.145E+01(5.955E-03)**
WFG3	1.085E+01(3.419E-02)-	1.092E+01(9.689E-03)-	1.092E+01(1.123E-02)-	1.094E+01(5.939E-03)-	**1.095E+01(9.340E-03)**
WFG4	7.928E+00(5.812E-01)-	8.319E+00(8.223E-02)-	8.356E+00(6.183E-02)-	8.646E+00(1.972E-02)-	**8.674E+00(5.515E-03)**
WFG5	7.946E+00(2.611E-02)-	8.095E+00(1.024E-02)-	8.102E+00(1.039E-02)-	8.135E+00(4.492E-03)-	**8.163E+00(3.597E-02)**
WFG6	8.422E+00(2.341E-01)~	**8.480E+00(1.180E-01)+**	8.320E+00(2.620E-01)~	8.458E+00(2.333E-01)~	8.389E+00(2.722E-01)
WFG7	8.352E+00(3.627E-01)-	8.654E+00(7.541E-03)-	8.660E+00(3.299E-03)-	8.679E+00(8.560E-04)-	**8.684E+00(8.940E-04)**
WFG8	6.855E+00(1.506E-01)-	7.008E+00(2.568E-01)-	**7.325E+00(5.059E-01)~**	7.043E+00(3.679E-02)~	7.169E+00(1.706E-01)
WFG9	7.681E+00(1.506E+00)-	8.144E+00(3.560E-02)-	7.901E+00(9.500E-01)-	7.994E+00(2.611E-01)-	**8.206E+00(4.384E-02)**
UF1	3.590E+00(7.199E-03)-	3.659E+00(2.229E-03)+	**3.660E+00(1.986E-03)+**	3.648E+00(9.914E-04)-	3.657E+00(7.212E-04)
UF2	3.085E+00(8.714E-01)-	3.640E+00(2.322E-02)-	3.650E+00(5.738E-03)~	3.638E+00(3.912E-02)-	**3.659E+00(7.242E-04)**
UF3	3.182E+00(7.247E-01)-	3.619E+00(2.959E-02)~	**3.660E+00(5.674E-03)+**	3.610E+00(7.880E-02)~	3.654E+00(7.457E-03)
UF4	3.073E+00(4.365E-02)-	3.158E+00(1.900E-02)-	3.166E+00(2.236E-02)-	**3.233E+00(4.725E-03)~**	3.232E+00(4.199E-03)
UF5	2.414E+00(2.039E-01)~	2.653E+00(2.998E-01)~	**2.703E+00(3.052E-01)~**	2.273E+00(5.117E-01)~	2.062E+00(3.064E+00)
UF6	2.947E+00(1.954E-01)~	2.843E+00(5.342E-01)~	**2.950E+00(3.425E-01)~**	2.720E+00(7.915E-01)~	2.591E+00(9.106E-01)
UF7	3.420E+00(9.423E-03)-	3.487E+00(1.505E-02)-	3.483E+00(1.042E-02)-	3.485E+00(1.279E-02)~	**3.495E+00(3.691E-04)**
UF8	**7.372E+00(1.067E-02)+**	7.320E+00(2.449E-02)-	7.315E+00(2.844E-02)-	7.270E+00(4.528E-03)-	7.355E+00(8.268E-03)
UF9	7.460E+00(4.836E-01)-	7.456E+00(4.341E-01)-	7.444E+00(4.530E-01)-	7.476E+00(5.108E-01)-	**7.571E+00(3.756E-01)**
UF10	3.409E+00(3.183E-01)-	3.444E+00(3.387E-01)-	3.576E+00(5.094E-01)-	5.043E+00(1.681E+00)-	**5.991E+00(3.414E-01)**
F1	3.442E+00(4.032E-01)-	3.665E+00(1.229E-04)~	3.665E+00(8.624E-05)+	**3.665E+00(8.115E-06)+**	3.665E+00(2.628E-05)
F2	3.356E+00(2.411E-01)-	3.410E+00(3.195E-01)-	3.643E+00(3.985E-03)-	3.595E+00(8.490E-03)-	**3.652E+00(6.979E-04)**
F3	3.406E+00(3.646E-01)-	3.575E+00(1.152E-01)-	**3.660E+00(2.343E-03)+**	3.654E+00(2.429E-03)-	3.659E+00(4.334E-04)
F4	3.471E+00(2.838E-02)-	3.628E+00(3.256E-02)-	3.661E+00(1.307E-03)+	**3.662E+00(1.049E-03)+**	3.658E+00(7.462E-04)
F5	3.515E+00(2.558E-01)-	3.620E+00(2.658E-02)-	**3.656E+00(1.807E-03)+**	3.648E+00(3.210E-03)-	3.654E+00(4.704E-03)
F6	**7.441E+00(1.369E-03)+**	7.421E+00(2.805E-03)-	7.421E+00(4.012E-03)-	7.416E+00(5.786E-03)-	7.429E+00(2.431E-03)
F7	3.517E+00(1.617E-01)-	3.567E+00(1.061E-01)-	3.654E+00(5.483E-03)+	**3.656E+00(7.562E-03)+**	3.636E+00(1.985E-02)
F8	3.227E+00(1.111E-01)-	3.424E+00(8.456E-02)-	**3.600E+00(4.207E-02)+**	3.513E+00(1.485E-01)~	3.489E+00(1.284E-01)
F9	3.131E+00(1.925E-01)-	3.113E+00(1.162E-01)-	**3.326E+00(1.144E-03)+**	3.220E+00(1.643E-03)-	3.316E+00(2.863E-03)
Best/All	2/35	1/35	9/35	6/35	17/35
Total	30-/3~/2+	29-/4~/2+	21-/5~/9+	22-/8~/5+	/

+: HMOPSO-ARA shows significantly better performance in the experimental comparison.
-: . HMOPSO-ARA shows significantly worse performance in the experimental comparison.
~: There is no significant difference between the compared experimental results.

HMOPSO-ARA outperforms MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD on 30, 29, 21 and 22 cases among all the 35 test problems, respectively according to the experimental results of Wilcoxon's rank sum test. While HMOPSO-ARA only underperforms MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD on 2, 2, 9 and 5 test problems out of all the 35 test problems, respectively. These HV results further validate that HMOPSO-ARA has some advantages when tackling all the test instances employed in this paper.

Some scatter plots of populations obtained by HMOPSO-ARA and peer algorithms are displayed below, which visually confirm the superiority of HMOPSO-ARA over other compared algorithms. As shown in Figs. 6-8 respectively showing the comparisons of approximation sets on UF3, F1 and F6, the final population obtained by HMOPSO-ARA can be distributed on the true ***PFs*** more smoothly, completely and closely, which have shown the obvious superiority of HMOPSO-ARA over other compared algorithms.

The above experimental results show that the proposed HMOPSO-ARA has some advantages in solving all the test problems applied in our experimental comparison, when comparing with four peer algorithms (i.e., MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD). The main reason for the obvious performance of HMOPSO-ARA is mainly benefited from the use of a novel velocity update function and an effective dynamic resource allocation strategy, which let

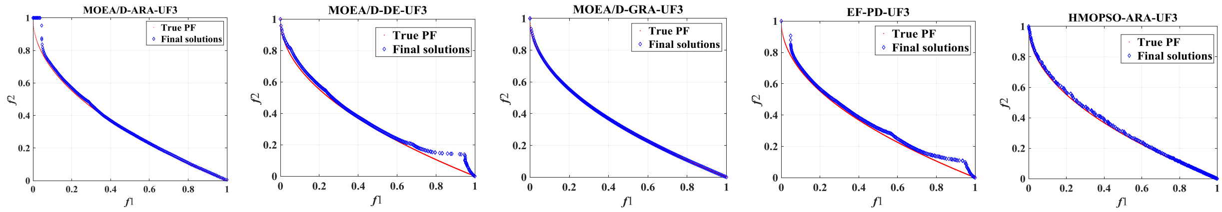


FIGURE 6. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on UF3.

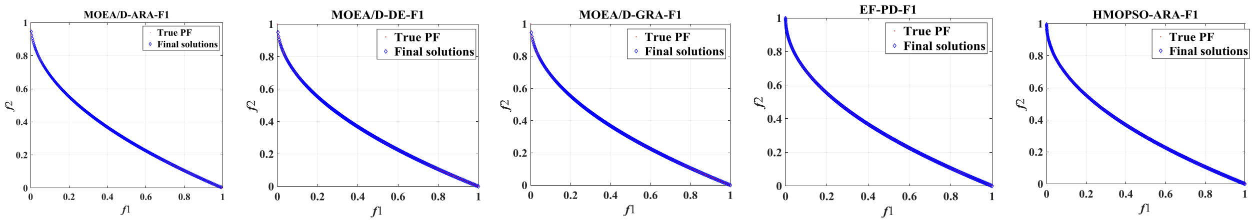


FIGURE 7. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on F1.

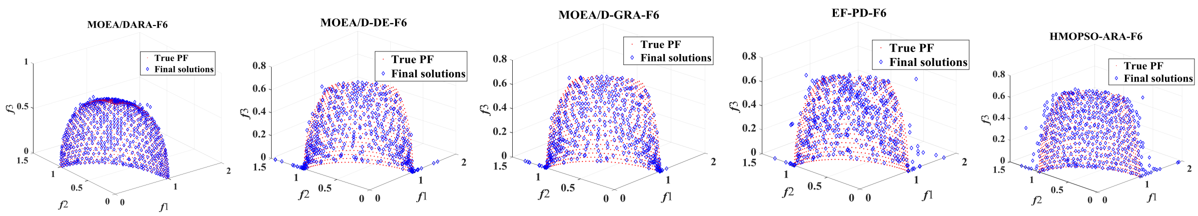


FIGURE 8. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on F6.

our proposed algorithm not only have the search pattern of MOPSOs, but also include the advantages of MOEAs based on the resource allocation strategy.

E. COMPARISON OF HMOPSO-ARA AND FOUR COMPETITIVE MOPSOs

In this subsection, four competitive MOPSOs (i.e., SMPSO [27], dMOPSO [29], CMPSO [30] and AgMOPSO [28]) are compared to further valid the effectiveness of our proposed algorithm HMOPSO-ARA. Tables 4 and 5 provide the IGD and HV results of all the compared algorithms on all the test problems, respectively. It is pointed out that all the experimental results were obtained after 30 independent runs.

As the IGD values summarized in Table 4, we can see that our proposed HMOPSO-ARA shows obvious effectiveness and advantages when compared with peer algorithms mentioned above when tackling all the applied test instances. In summary, HMOPSO-ARA shows the best performance on 28 cases among all the 35 test cases, while the compared algorithms (i.e., SMPSO, CMPSO, dMOPSO and AgMOPSO) can only respectively obtain the best performance on 0, 2, 1 and 4 cases among all the 35 cases,. Furthermore, HMOPSO-ARA respectively outperforms peer algorithms on 32, 28, 33 and 26 test problems based on the Wilcoxon's rank sum test, while it respectively underperforms SMPSO, CMPSO, dMOPSO and AgMOPSO on 1, 2, 1 and 4 cases among all the 35 cases. Hence, some conclusions can be drawn from the experimental results listed in Table 4, and the proposed HMOPSO-ARA shows obvious advantages over the peer algorithms (i.e., SMPSO, CMPSO, dMOPSO and AgMOPSO). As all the mentioned algorithms are proposed based on the PSO-based search method, the most difference between HMOPSO-ARA and the other compared algorithms are the adaptive resource allocation strategy designed in HMOPSO-ARA, which plays a crucial role in tackling some complicated MOPs (i.e., UF1-UF10 and F1-F9 test problems). Hence, when compared with other traditional MOPSOs, our proposed HMOPSO-ARA employing the resource allocation strategy into the traditional MOPSOs shows obvious advantages over peer algorithms. By this way, the computational resources will not be allocated to all the particles equally. On the contrary, most of computational resources will be allocated to these particles with better performance, which can enhance the performance and robustness of our algorithm when solving some complicated MOPs (i.e., UF1-UF10 and F1-F9 test problems).

As observed from the HV values of HMOPSO-ARA and peer algorithms on solving all the 35 test problems (i.e., DTLZ1-DTLZ7, WFG1-WFG9, UF1-UF10, and F1-F9) in Table 5, some conclusions can be obtained from

TABLE 4. Experimental results (mean and standard deviation) of HV values on all test problems obtained by SMPSO, CMPSO, dMOPSO, AgMOPSO and HMOPSO-ARA.

Instances	SMPSO	CMPSO	dMOPSO	AgMOPSO	HMOPSO-ARA
DTLZ1	3.072E-02(3.835E-03)-	1.441E-01(8.031E-02)-	3.475E-02(5.608E-03)-	4.794E-02(1.465E-03)-	**1.990E-02(6.641E-04)**
DTLZ2	6.965E-02(3.447E-03)-	6.308E-02(3.239E-03)-	6.789E-02(6.304E-04)-	**5.176E-02(6.258E-04)+**	5.236E-02(7.782E-04)
DTLZ3	1.206E-01(1.205E-02)-	3.347E-01(2.632E-01)-	7.925E-02(2.766E-03)-	1.019E-01(1.413E-03)-	**5.290E-02(7.662E-04)**
DTLZ4	5.795E-02(1.985E-02)-	5.817E-02(1.455E-02)~	5.702E-02(2.287E-03)-	**3.781E-02(2.294E-03)~**	3.895E-02(2.739E-03)
DTLZ5	4.060E-03(1.683E-04)-	5.367E-03(3.084E-04)-	1.262E-02(2.372E-04)-	**3.930E-03(1.211E-04)~**	3.961E-03(1.592E-04)
DTLZ6	3.956E-03(2.157E-04)-	4.502E-03(1.799E-04)-	1.194E-02(3.373E-06)-	3.801E-03(1.549E-04)~	**3.737E-03(2.186E-04)**
DTLZ7	8.572E-02(8.511E-03)-	7.232E-02(4.420E-03)-	1.983E-01(1.518E-03)-	**5.752E-02(1.583E-03)+**	5.826E-02(1.901E-03)
WFG1	1.212E+00(1.121E-02)-	8.193E-01(2.019E-01)-	1.199E+00(7.213E-03)-	2.421E-01(9.909E-02)-	**1.235E-02(2.634E-04)**
WFG2	2.776E-02(9.696E-03)-	6.207E-02(1.638E-03)-	5.837E-02(1.229E-02)-	1.065E-02(8.581E-04)-	**1.019E-02(2.805E-04)**
WFG3	1.943E-02(1.881E-03)-	1.529E-02(1.607E-03)-	2.719E-02(3.373E-03)-	1.205E-02(3.486E-04)-	**1.179E-02(4.199E-04)**
WFG4	6.200E-02(4.098E-03)-	1.304E-02(5.234E-04)-	3.705E-02(7.254E-03)-	1.215E-02(1.291E-03)-	**1.075E-02(2.992E-04)**
WFG5	6.666E-02(2.223E-04)-	**6.577E-02(2.762E-03)~**	6.711E-02(2.148E-04)-	6.630E-02(6.680E-05)-	6.585E-02(8.294E-05)
WFG6	4.553E-02(4.989E-02)~	2.790E-02(1.273E-02)+	**2.301E-02(3.249E-03)+**	3.296E-02(3.085E-02)+	5.039E-02(4.134E-02)
WFG7	1.631E-02(1.735E-03)-	1.515E-02(1.010E-03)-	2.549E-02(2.236E-03)-	1.215E-02(1.864E-04)~	**1.206E-02(2.231E-04)**
WFG8	2.381E-01(2.718E-02)-	2.069E-01(2.124E-02)~	2.493E-01(2.367E-02)-	2.323E-01(1.419E-02)-	**2.048E-01(2.021E-02)**
WFG9	2.248E-02(1.065E-03)+	**2.143E-02(3.645E-03)+**	2.438E-02(1.850E-03)+	3.308E-02(1.996E-03)~	4.466E-02(5.105E-03)
UF1	1.748E-01(6.267E-02)-	5.870E-02(2.305E-02)-	5.875E-02(9.627E-03)-	1.290E-02(1.770E-03)-	**5.679E-03(4.385E-04)**
UF2	5.940E-02(1.048E-02)-	2.413E-02(9.265E-03)-	1.879E-02(1.651E-03)-	1.037E-02(9.729E-04)-	**5.008E-03(4.866E-04)**
UF3	2.735E-01(3.153E-02)-	1.008E-01(1.705E-02)-	1.329E-01(5.905E-02)-	3.974E-02(3.340E-02)-	**7.594E-03(4.383E-03)**
UF4	6.192E-02(7.131E-03)-	4.437E-02(2.923E-03)-	4.695E-02(2.432E-03)-	4.020E-02(7.751E-04)-	**3.894E-02(6.673E-04)**
UF5	2.200E+00(1.038E+00)-	2.849E-01(6.868E-02)~	1.127E+00(3.603E-01)-	4.057E-01(2.384E-01)-	**2.666E-01(2.353E-01)**
UF6	8.556E-01(2.184E-01)-	3.307E-01(7.877E-02)~	5.205E-01(1.112E-01)-	3.365E-01(2.081E-01)~	**2.838E-01(3.349E-01)**
UF7	9.360E-02(2.301E-02)-	7.507E-02(1.828E-02)-	2.300E-02(2.508E-03)-	1.113E-02(4.499E-03)-	**2.905E-03(2.045E-04)**
UF8	2.142E-01(1.416E-02)-	3.340E-01(1.125E-01)-	2.901E-01(6.221E-02)-	7.708E-02(1.723E-03)-	**5.685E-02(3.116E-03)**
UF9	4.159E-01(3.055E-02)-	2.595E-01(5.356E-02)-	1.482E-01(7.057E-02)-	1.503E-01(1.466E-02)-	**7.007E-02(7.209E-02)**
UF10	2.386E-01(4.227E-02)~	4.749E-01(1.789E-01)-	4.149E-01(8.043E-02)-	4.208E-01(1.268E-01)-	**2.411E-01(2.211E-02)**
F1	5.325E-03(8.047E-04)-	4.176E-03(1.288E-04)-	3.024E-03(2.443E-04)-	1.352E-03(2.725E-05)-	**1.272E-03(2.417E-05)**
F2	2.297E-01(5.824E-02)-	9.171E-02(9.521E-03)-	1.054E-01(6.577E-03)-	3.815E-02(1.024E-02)-	**8.668E-03(3.165E-04)**
F3	1.320E-01(1.620E-02)-	4.176E-02(9.065E-03)-	7.245E-02(1.246E-02)-	1.741E-02(2.665E-03)-	**4.946E-03(3.484E-04)**
F4	1.262E-01(7.174E-02)-	2.787E-02(1.326E-02)-	5.851E-02(3.994E-03)-	1.919E-02(1.742E-03)-	**5.879E-03(6.094E-04)**
F5	9.508E-02(1.511E-02)-	3.156E-02(6.444E-03)-	6.058E-02(7.063E-03)-	1.601E-02(1.632E-03)-	**6.640E-03(3.709E-04)**
F6	3.005E+00(1.881E+00)-	1.185E-01(1.832E-02)-	7.748E-02(9.612E-03)-	3.719E-02(4.651E-03)-	**2.955E-02(1.022E-03)**
F7	3.270E-01(8.410E-02)-	6.860E-02(2.531E-02)-	3.271E-01(3.145E-02)-	1.620E-02(1.421E-02)-	**3.645E-03(9.133E-04)**
F8	3.536E-01(3.331E-02)-	1.617E-01(6.247E-02)-	3.665E-01(6.363E-03)-	1.414E-01(9.670E-02)-	**4.099E-02(1.947E-02)**
F9	2.517E-01(6.555E-02)-	7.829E-02(1.353E-02)-	1.162E-01(7.443E-03)-	4.812E-02(1.607E-02)-	**1.401E-02(3.816E-03)**
Best/All	0/35	2/35	1/35	4/35	28/35
Total	32-/2~/1+	28-/5~/2+	33-/0~/2+	26-/6~/3+	/

+: HMOPSO-ARA shows significantly better performance in the experimental comparison.
-: . HMOPSO-ARA shows significantly worse performance in the experimental comparison.
~: There is no significant difference between the compared experimental results.

the experimental results that our proposed HMOPSO-ARA can outperform significantly on most test problems, especially on UF1-UF10 and F1-F9 test problems with variable linkages. Totally, HMOPSO-ARA can obtain 29 best cases among all the 35 cases, while SMPSO, CMPSO, dMOPSO and AgMOPSO can only perform best on 1, 1, 1 and 3 cases, respectively. Furthermore, according to the Wilcoxon's rank sum test listed in the last row of Table 5, our proposed HMOPSO-ARA can respectively outperform SMPSO, CMPSO, dMOPSO and AgMOPSO on 31, 31, 33 and 26 test instances among all the 35 test suits. However, HMOPSO-ARA just underperforms SMPSO, CMPSO, dMOPSO and AgMOPSO on 1, 2, 2 and 4 cases among all the 35 test cases, respectively. Moreover, it should be pointed out that our proposed algorithm HMOPSO-ARA shows obvious superiority over other traditional MOPSOs on tackling some complicated MOPs with variable linkages (i.e., UF1-UF10 and F1-F9 test problems). According to the HV results on UF and F test problems, the proposed HMOPSO-ARA can obtain the best results on almost all the UF and F test problems except UF5 and UF10, as AgMOPSO can obtain the best results on UF5 test instance and dMOPSO outperforms on UF10 test problem, respectively. Hence, the effectiveness of our proposed adaptive resource allocation strategy employed in MOPSO is further validated. In summary, the proposed HMOPSO-ARA has shown an obvious superiority over other traditional MOPSOs according to the HV results listed in Table 5.

TABLE 5. Experimental results (mean and standard deviation) of IGD and HV values on all test problems obtained by HMOPSO-ARA and HMOPSO.

Instances	SMPSO	CMPSO	dMOPSO	AgMOPSO	HMOPSO-ARA
DTLZ1	9.549E-01(2.280E-03)-	8.944E-01(4.872E-02)-	9.584E-01(1.490E-02)-	9.630E-01(6.997E-04)-	**9.735E-01(2.426E-04)**
DTLZ2	7.353E+00(1.322E-02)-	7.385E+00(5.487E-03)-	7.385E+00(2.076E-03)-	**7.415E+00(1.473E-03)+**	7.414E+00(1.062E-03)
DTLZ3	6.846E+00(1.948E-01)-	6.775E+00(5.610E-01)-	7.360E+00(6.346E-03)-	7.203E+00(5.596E-03)-	**7.413E+00(2.697E-03)**
DTLZ4	7.363E+00(1.628E-02)-	7.180E+00(1.829E-01)-	7.314E+00(2.778E-02)-	7.415E+00(1.030E-03)~	**7.416E+00(1.471E-03)**
DTLZ5	2.210E+01(4.286E-04)~	2.210E+01(6.603E-04)-	2.209E+01(1.942E-04)-	**2.210E+01(1.191E-04)+**	2.210E+01(3.921E-04)
DTLZ6	2.210E+01(1.348E-04)-	2.210E+01(2.895E-04)-	2.209E+01(1.407E-05)-	2.210E+01(1.126E-04)~	**2.210E+01(4.409E-05)**
DTLZ7	9.237E+00(7.407E-02)-	9.331E+00(4.881E-02)-	9.271E+00(5.716E-03)-	9.477E+00(1.492E-02)~	**9.478E+00(1.291E-02)**
WFG1	5.434E+00(2.618E-02)-	7.378E+00(1.031E+00)-	5.475E+00(3.779E-02)-	1.059E+01(5.987E-01)-	**1.207E+01(3.312E-04)**
WFG2	1.131E+01(6.877E-02)-	1.068E+01(1.794E-02)-	1.124E+01(8.981E-02)-	1.145E+01(4.283E-03)-	**1.145E+01(5.955E-03)**
WFG3	1.087E+01(1.714E-02)-	1.092E+01(1.611E-02)-	1.082E+01(3.027E-02)-	1.094E+01(5.036E-03)-	**1.095E+01(9.340E-03)**
WFG4	8.182E+00(3.981E-02)-	8.647E+00(1.080E-02)-	8.427E+00(6.081E-02)-	8.652E+00(1.205E-02)-	**8.674E+00(5.515E-03)**
WFG5	8.000E+00(6.825E-02)-	8.109E+00(2.530E-02)-	8.099E+00(2.263E-02)-	8.146E+00(2.324E-02)-	**8.163E+00(3.597E-02)**
WFG6	8.418E+00(3.332E-01)~	8.546E+00(1.019E-01)+	**8.572E+00(3.840E-02)+**	8.516E+00(2.312E-01)+	8.389E+00(2.722E-01)
WFG7	8.627E+00(1.403E-02)-	8.667E+00(3.248E-03)-	8.558E+00(4.208E-02)-	8.680E+00(1.784E-03)-	**8.684E+00(8.940E-04)**
WFG8	6.975E+00(3.683E-01)-	7.140E+00(1.545E-01)~	6.900E+00(2.109E-01)-	6.981E+00(1.094E-01)-	**7.169E+00(1.706E-01)**
WFG9	8.219E+00(6.793E-02)+	**8.333E+00(2.853E-02)+**	8.264E+00(5.905E-02)+	8.263E+00(2.253E-02)+	8.206E+00(4.384E-02)
UF1	3.207E+00(2.064E-01)-	3.495E+00(1.289E-01)-	3.500E+00(8.061E-02)-	3.645E+00(3.385E-03)-	**3.657E+00(7.212E-04)**
UF2	3.496E+00(3.296E-02)-	3.599E+00(7.297E-02)-	3.621E+00(1.556E-02)-	3.645E+00(8.141E-03)-	**3.659E+00(7.242E-04)**
UF3	3.172E+00(4.695E-02)-	3.486E+00(2.672E-02)-	3.475E+00(8.345E-02)-	3.599E+00(5.361E-02)-	**3.654E+00(7.457E-03)**
UF4	3.165E+00(2.449E-02)-	3.207E+00(6.144E-03)-	3.207E+00(5.488E-03)-	3.226E+00(2.850E-03)-	**3.232E+00(4.199E-03)**
UF5	8.545E-02(4.537E-02)-	2.213E+00(4.048E-01)~	7.344E-01(5.871E-01)-	**2.290E+00(4.097E-01)~**	2.062E+00(3.064E+00)
UF6	1.248E+00(3.567E-01)-	1.680E+00(2.516E+00)-	1.965E+00(3.762E-01)-	2.468E+00(4.894E-01)~	**2.591E+00(9.106E-01)**
UF7	3.214E+00(6.479E-02)-	3.346E+00(4.064E-02)-	3.435E+00(1.255E-02)-	3.462E+00(3.825E-02)-	**3.495E+00(3.691E-04)**
UF8	6.335E+00(3.408E-02)-	6.482E+00(1.059E-01)-	6.514E+00(1.122E-04)-	7.273E+00(1.047E-02)-	**7.355E+00(8.268E-03)**
UF9	5.136E+00(2.018E-01)-	6.647E+00(6.016E-01)-	7.088E+00(1.959E-01)-	7.117E+00(9.671E-02)-	**7.571E+00(3.756E-01)**
UF10	**6.137E+00(2.870E-01)~**	4.439E+00(2.226E+00)-	5.941E+00(1.670E-04)-	3.962E+00(1.021E+00)-	5.991E+00(3.414E-01)
F1	3.657E+00(1.499E-03)-	3.660E+00(2.912E-04)-	3.661E+00(4.885E-04)-	3.664E+00(7.634E-05)-	**3.665E+00(2.628E-05)**
F2	2.792E+00(2.864E-01)-	3.328E+00(3.088E-02)-	3.182E+00(2.696E-02)-	3.545E+00(1.470E-01)-	**3.652E+00(6.979E-04)**
F3	3.279E+00(5.631E-02)-	3.514E+00(1.495E-02)-	3.439E+00(5.832E-02)-	3.637E+00(4.760E-03)-	**3.659E+00(4.334E-04)**
F4	3.317E+00(2.867E-01)-	3.625E+00(1.200E-02)-	3.539E+00(1.479E-02)-	3.633E+00(3.206E-03)-	**3.658E+00(7.462E-04)**
F5	3.359E+00(3.600E-02)-	3.570E+00(8.565E-02)-	3.467E+00(5.100E-02)-	3.636E+00(1.088E-02)-	**3.654E+00(4.704E-03)**
F6	1.233E-01(0.000E+00)-	7.181E+00(1.971E-01)-	7.307E+00(2.323E-02)-	7.406E+00(1.117E-02)-	**7.429E+00(2.431E-03)**
F7	2.961E+00(8.277E-02)-	3.512E+00(3.153E-02)-	2.828E+00(6.129E-01)-	3.553E+00(8.308E-02)-	**3.636E+00(1.985E-02)**
F8	3.042E+00(5.577E-02)-	3.177E+00(3.713E-01)-	3.005E+00(3.067E-02)-	3.201E+00(3.459E-01)-	**3.489E+00(1.284E-01)**
F9	2.338E+00(1.114E-01)-	2.992E+00(1.329E-01)-	2.660E+00(2.875E-02)-	3.206E+00(5.822E-02)-	**3.316E+00(2.863E-03)**
Best/All	1/35	1/35	1/35	3/35	29/35
Total	31-/3~/1+	31-/2~/2+	33-/0~/2+	26-/5~/4+	/

+: HMOPSO-ARA shows significantly better performance in the experimental comparison.
-: . HMOPSO-ARA shows significantly worse performance in the experimental comparison.
~: There is no significant difference between the compared experimental results.

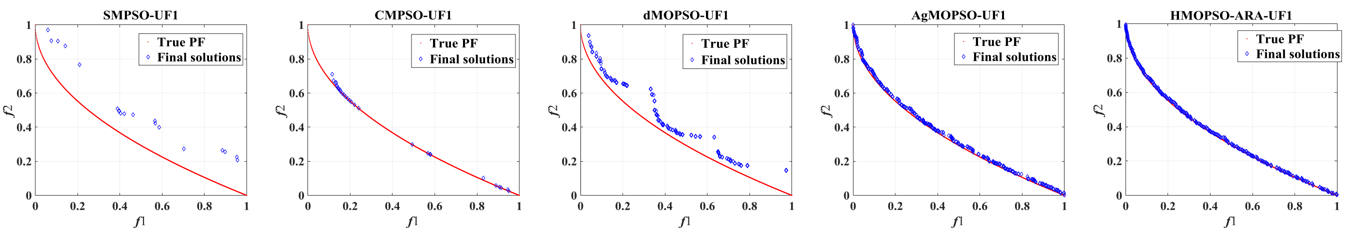


FIGURE 9. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on UF1.

In order to have a visual observation about the superior performance of HMOPSO-ARA, the final solutions obtained by HMOPSO-ARA and other compared algorithms (i.e., SMPSO, CMPSO, dMOPSO and AgMOPSO) are plotted below. The scatter plots of populations on UF1, UF3, F2 and F3 test problems are respectively drawn in Figs. 9-12, where blue rhombus and red dots respectively indicate the final solutions and the true ***PFs***. It is easily learned from these figures that SMPSO, CMPSO, dMOPSO and AgMOPSO have failed in finding a whole population, which can closely

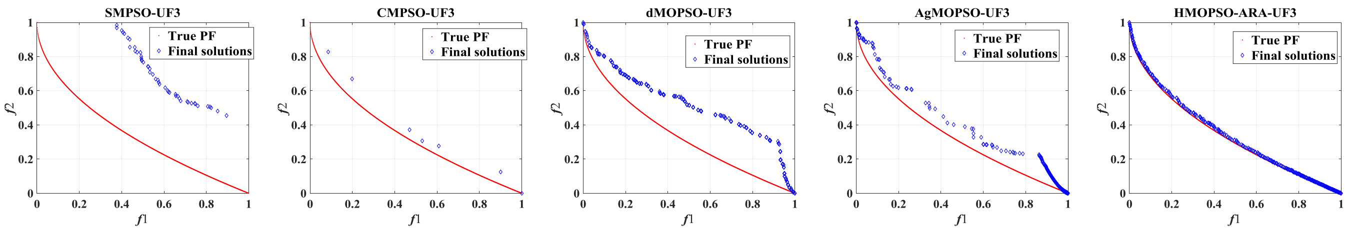


FIGURE 10. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on UF3.

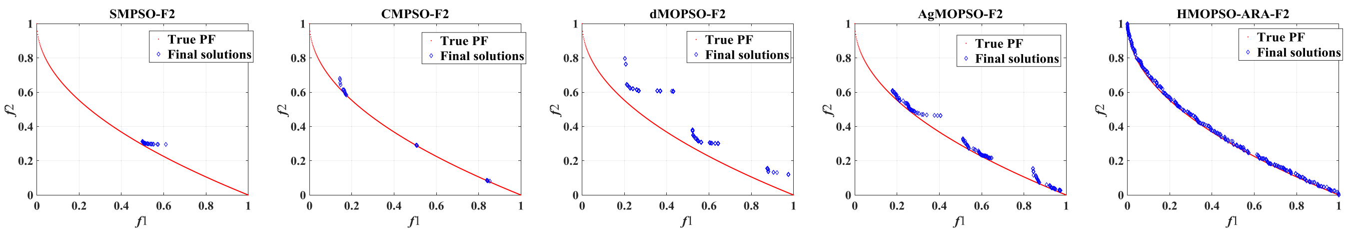


FIGURE 11. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on F2.

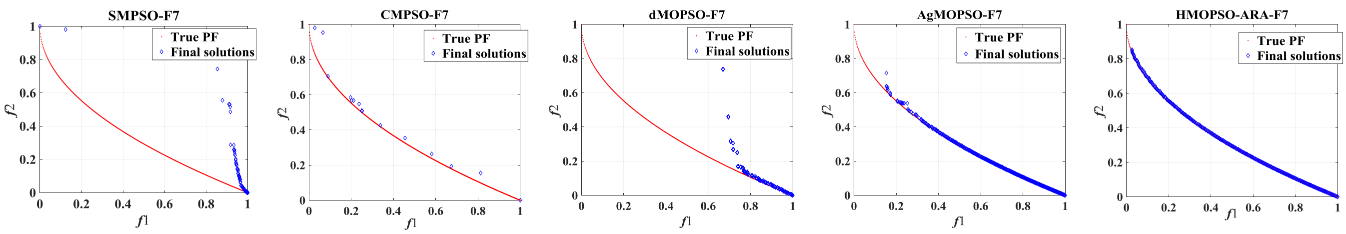


FIGURE 12. Scatter plots of populations obtained by HMOPSO-ARA and the compared algorithms on F7.

and evenly cover the true ***PFs***. However, the proposed HMOPSO-ARA shows an obvious superiority over the competitors, as its final population can cover the entire true ***PFs*** evenly and completely.

F. A FURTHER DISCUSSION OF ADAPTIVE RESOURCE ALLOCATION STRATEGY

In this subsection, the advantages of employing the proposed adaptive resource allocation strategy into tradition MOPSOs is further discussed. HMOPSO-ARA is compared with its variant without using the proposed adaptive resource allocation strategy, called HMOPSO, in which the computation resources will be allocated equally for each particle. To be fair, the relative parameters in HMOPSO are set according to the parameters shown in Table 1. The IGD and HV values of HMOPSO-ARA and HMOPSO are summarized in Table 6, which further validate the effectiveness of our proposed adaptive resource allocation strategy employed into traditional MOPSOs. It should be known that all the experimental results are obtained after running 30 independent times.

According to the IGD results listed in Table 6, the proposed HMOPSO-ARA shows some superiority over HMOPSO, which is mainly benefited from the idea that combines the adaptive resource allocation strategy with traditional MOPSOs. HMOPSO-ARA can perform best on 26 cases among all the 35 cases, while HMOPSO can only show the best performance on 9 test cases out of all the 35 test cases. As shown in the last row of Table 6, the proposed HMOPSO-ARA outperforms HMOPSO on 14 test problems out of all the 35 test instances. On the contrary, HMOPSO can only outperform HMOPSO-ARA on one test problem when considering the IGD values. Similar conclusions can be drawn from the HV results summarized in Table 6, as the proposed HMOPSO-ARA can obtain the best performance on 26 cases out of all the 35 cases, while HMOPSO without the adaptive resource allocation can only show the best performance on 9 test problems among all the 35 test problems. Moreover, the proposed HMOPSO-ARA can outperform HMOPSO on 15 test problems, but only underperforms HMOPSO on one test problem among all the 35 test problems. Hence, the effectiveness of our proposed adaptive resource allocation strategy in MOPSOs and the superiority of our proposed algorithm have been further validated according to the experimental results.

G. A TIME COMPLEXITY ANALYSIS OF HMOPSO-ARA

When analyzing the computational complexity of the proposed HMOPSO-ARA, the main procedure in one generation is the main loop of **Algorithm 1**. As illustrated above, there

TABLE 6. Experimental results (mean and standard deviation) of IGD and HV values on all test problems obtained by HMOPSO-ARA and HMOPSO.

Instances	IGD values		HV values	
	HMOPSO	HMOPSO-ARA	HMOPSO	HMOPSO-ARA
DTLZ1	2.067E-02(5.325E-04)-	**1.990E-02(6.641E-04)**	9.731E-01(1.363E-04)-	**9.735E-01(2.426E-04)**
DTLZ2	5.272E-02(7.768E-04)~	**5.236E-02(7.782E-04)**	7.414E+00(1.911E-03)-	**7.414E+00(1.062E-03)**
DTLZ3	5.399E-02(1.031E-03)~	**5.290E-02(7.662E-04)**	7.409E+00(3.649E-03)-	**7.413E+00(2.697E-03)**
DTLZ4	5.730E-02(2.553E-03)-	**3.895E-02(2.739E-03)**	7.044E+00(1.977E-03)-	**7.416E+00(1.471E-03)**
DTLZ5	4.282E-03(1.617E-04)-	**3.961E-03(1.592E-04)**	2.210E+01(7.369E-04)-	**2.210E+01(3.921E-04)**
DTLZ6	3.746E-03(1.394E-04)~	**3.737E-03(2.186E-04)**	2.210E+01(1.032E-04)~	**2.210E+01(4.409E-05)**
DTLZ7	**5.792E-02(1.212E-03)~**	5.826E-02(1.901E-03)	**9.480E+00(2.248E-02)~**	9.478E+00(1.291E-02)
WFG1	1.786E-02(3.394E-03)~	**1.235E-02(2.634E-04)**	1.204E+01(5.479E-03)~	**1.207E+01(3.312E-04)**
WFG2	1.025E-02(5.556E-04)~	**1.019E-02(2.805E-04)**	**1.145E+01(4.205E-03)~**	1.145E+01(5.955E-03)
WFG3	1.190E-02(2.895E-04)~	**1.179E-02(4.199E-04)**	1.094E+01(6.193E-03)~	**1.095E+01(9.340E-03)**
WFG4	**1.068E-02(2.147E-04)~**	1.075E-02(2.992E-04)	**8.674E+00(4.823E-03)~**	8.674E+00(5.515E-03)
WFG5	6.626E-02(3.859E-05)~	**6.585E-02(8.294E-05)**	**8.166E+00(3.674E-02)~**	8.163E+00(3.597E-02)
WFG6	5.891E-02(6.862E-02)-	**5.039E-02(4.134E-02)**	8.339E+00(4.585E-01)~	**8.389E+00(2.722E-01)**
WFG7	**1.196E-02(3.042E-04)~**	1.206E-02(2.231E-04)	8.684E+00(1.026E-03)-	**8.684E+00(8.940E-04)**
WFG8	2.284E-01(1.168E-02)-	**2.048E-01(2.021E-02)**	6.985E+00(7.696E-02)-	**7.169E+00(1.706E-01)**
WFG9	**3.422E-02(5.551E-03)~**	4.466E-02(5.105E-03)	**8.252E+00(5.947E-02)~**	8.206E+00(4.384E-02)
UF1	**5.610E-03(7.125E-04)~**	5.679E-03(4.385E-04)	3.657E+00(1.048E-03)~	**3.657E+00(7.212E-04)**
UF2	**4.644E-03(4.999E-04)+**	5.008E-03(4.866E-04)	3.658E+00(5.599E-04)~	**3.659E+00(7.242E-04)**
UF3	1.084E-02(7.740E-03)-	**7.594E-03(4.383E-03)**	3.649E+00(1.303E-02)~	**3.654E+00(7.457E-03)**
UF4	3.911E-02(4.327E-04)~	**3.894E-02(6.673E-04)**	3.231E+00(5.281E-03)~	**3.232E+00(4.199E-03)**
UF5	**2.218E-01(1.347E-01)~**	2.666E-01(2.353E-01)	**2.123E+00(3.078E+00)~**	2.062E+00(3.064E+00)
UF6	3.322E-01(1.973E-01)~	**2.838E-01(3.349E-01)**	**2.603E+00(5.116E-01)~**	2.591E+00(9.106E-01)
UF7	3.282E-03(3.736E-04)-	**2.905E-03(2.045E-04)**	3.494E+00(8.327E-04)-	**3.495E+00(3.691E-04)**
UF8	6.429E-02(1.014E-02)-	**5.685E-02(3.116E-03)**	7.302E+00(2.893E-02)-	**7.355E+00(8.268E-03)**
UF9	9.349E-02(1.145E-01)-	**7.007E-02(7.209E-02)**	7.440E+00(5.476E-01)~	**7.571E+00(3.756E-01)**
UF10	**2.385E-01(2.901E-02)~**	2.411E-01(2.211E-02)	5.792E+00(3.669E-01)~	**5.991E+00(3.414E-01)**
F1	**1.269E-03(2.193E-05)~**	1.272E-03(2.417E-05)	**3.665E+00(2.351E-05)+**	3.665E+00(2.628E-05)
F2	9.805E-03(7.696E-04)-	**8.668E-03(3.165E-04)**	3.650E+00(1.236E-03)-	**3.652E+00(6.979E-04)**
F3	5.053E-03(3.592E-04)~	**4.946E-03(3.484E-04)**	**3.659E+00(4.197E-04)~**	3.659E+00(4.334E-04)
F4	6.679E-03(4.519E-04)-	**5.879E-03(6.094E-04)**	3.656E+00(5.843E-04)-	**3.658E+00(7.462E-04)**
F5	6.873E-03(7.668E-04)~	**6.640E-03(3.709E-04)**	3.653E+00(6.123E-03)~	**3.654E+00(4.704E-03)**
F6	3.053E-02(2.595E-03)-	**2.955E-02(1.022E-03)**	7.427E+00(5.345E-03)-	**7.429E+00(2.431E-03)**
F7	4.428E-03(1.917E-03)-	**3.645E-03(9.133E-04)**	3.623E+00(2.623E-02)-	**3.636E+00(1.985E-02)**
F8	4.531E-02(2.274E-02)-	**4.099E-02(1.947E-02)**	3.440E+00(1.568E-01)-	**3.489E+00(1.284E-01)**
F9	1.646E-02(7.108E-03)~	**1.401E-02(3.816E-03)**	3.313E+00(6.646E-03)-	**3.316E+00(2.863E-03)**
Best/All	9/35	26/35	9/35	26/35
Total	14-/20~/1+	/	15-/19~/1+	/

+: HMOPSO-ARA shows significantly better performance in the experimental comparison.
-: . HMOPSO-ARA shows significantly worse performance in the experimental comparison.
~: There is no significant difference between the compared experimental results.

are three main parts in the whole evolutionary process, i.e., archive-based evolutionary search process, PSO-based search process and archive update produce.

The computational complexity of each component is analyzed below. For the PSO-based search part as shown in **Algorithm 3**, its time complexity for velocity function update is $O(MN)$, where M is the objective number and N is the population size. Regarding to the evolutionary search section as introduced in **Algorithm 4**, the time complexity for cloning process, SBX and PM operators is $O(MN)$. When considering the time complexity of archive update as illustrated in **Algorithm 5**, the time complexity is $O(MN^2)$.

In summary, the worst case overall computational complexity of our proposed HMOPSO-ARA within one generation is $O(MN^2)$, which shows that HMOPSO-ARA is computationally efficient.

V. CONCLUSION AND FUTURE WORKS

In this paper, a novel hybrid multi-objective particle swarm optimization algorithm that proposes an adaptive resource allocation strategy into traditional MOPSO algorithm, is presented, namely HMOPSO-ARA. During the dynamic resource allocation process, more computation resource will be allocated for these particles with better performance on fitness improvement. Moreover, an archive-based evolutionary search with a decomposition-based clonal selection strategy is also performed on the external archive. Besides that, a novel velocity update function with another search direction is presented during the PSO-based search process in our algorithm to further speed up the convergence and maintain the diversity. By this way, our proposed HMOPSO-ARA has a strong ability to solve all the test problems adopted in our experimental comparison. When compared to four state-of-the-art MOEAs (i.e., MOEA/D-ARA, MOEA/D-DE, MOEA/D-GRA and EF_PD) and four competitive MOPSOs (i.e., SMPSO, CMPSO, dMOPSO and AgMOPSO), the experimental comparisons have further validated the advantages and effectiveness of our proposed HMOPSO-ARA on solving various kinds of MOPs, especially for some complicated test problems with variable linkages, such as UF1-UF10 and F1-F9 test problems.

Furthermore, we plan to extend our algorithm to solve many-objective optimization problems in our future work and the application of HMOPSO-ARA in tackling some real-word engineering problems is also a further research point.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.

[2] X. Li, W. Luo, and P. Xu, "Differential evolution for multimodal optimization with species by nearest-better clustering," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2019.2907657.

[3] Y. Q. Liu, H. Qin, Z. D. Zhang, L. Q. Yao, C. Wang, L. Mo, S. Ouyang, and J. Li, "A region search evolutionary algorithm for many-objective optimization," *Inf. Sci.*, vol. 488, pp. 19–40, Jul. 2019.

[4] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Lei, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, Apr. 2016.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab., Swiss Federal Inst. Technol. (ETH), Zürich, Switzerland, TIK Rep. 103, 2001, p. 103.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] Y. Liu, N. Zhu, and K. Li, "An angle dominance criterion for evolutionary many-objective optimization," *Inf. Sci.*, to be published, doi: 10.1016/j.ins.2018.12.078.

[8] Z. Cui, J. Zhang, Y. Chang, X. Cai, and W. Zhang, "Improved NSGA-III with selection-and-elimination operator," *Swarm Evol. Comput.*, vol. 49, pp. 23–33, Sep. 2019.

[9] J. Yi, J. Bai, H. He, J. Peng, and D. Tang, "ar-MOEA: A novel preference-based dominance relation for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 788–802, Oct. 2019, doi: 10.1109/TEVC.2018.2884133.

[10] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[11] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 203–208.

[12] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, Feb. 2016.

[13] Q. Lin, G. M. Jin, Y. P. Ma, K. C. Wong, C. A. C. Coello, J. Q. Li, J. Y. Chen, and J. Zhang, "A diversity-enhanced resource allocation strategy for decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2388–2401, Aug. 2018.

[14] Q. Kang, X. Song, M. Zhou, and L. Li, "A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2416–2423, Dec. 2019, doi: 10.1109/TSMC.2018.2818175.

[15] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, Apr. 2018.

[16] M. Wu, K. Li, S. Kwon, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2872803.

[17] D. Han, W. Du, W. Du, Y. Jin, and C. Wu, "An adaptive decomposition-based evolutionary algorithm for many-objective optimization," *Inf. Sci.*, vol. 491, pp. 204–222, Jul. 2019.

[18] D. H. Phan and J. Suzuki, "R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1836–1845.

[19] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[20] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature* (Lecture Notes in Computer Science), vol. 3242. Berlin, Germany: Springer, 2004, pp. 832–842.

[21] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, pp. 1653–1669, Sep. 2007.

[22] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, and X. Yao, "A scalable indicator-based evolutionary algorithm for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525–537, Jun. 2019, doi: 10.1109/TEVC.2018.2881153.

[23] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.

[24] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[25] G. Xu, Q. L. Cui, X. H. Shi, H. W. Ge, Z. H. Zhan, H. P. Lee, Y. C. Liang, R. Tai, and C. G. Wu, "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.*, vol. 45, pp. 33–51, Mar. 2019.

[26] X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, H. Wu, and Z.-H Zhan, "An expanded particle swarm optimization based on multi-exemplar and forfetting ability," *Inf. Sci.*, vol. 508, pp. 105–120, Jan. 2020.

[27] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. C. Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Multi-Criteria Decis.-Making*, Mar./Apr. 2009, pp. 66–73.

[28] Q. Zhu, Q. Lin, W. Chen, K.-C. Wong, and A. C. C. Coello, "An external archive-guided multiobjective particle swarm optimization algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2794–2808, Sep. 2017.

[29] S. Z. Martínez and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, Jul. 2011, pp. 69–76.

[30] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[31] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization* (Advanced Information and Knowledge Processing), A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[33] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* in Evolutionary Multi-Criterion Optimization, vol. 3410, C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds. Guanajuato, Mexico: Springer-Verlag, 2005, pp. 280–295.

[34] W. Peng, L. Bo, and Z. Wen, "Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D," *Appl. Soft Comput.*, vol. 70, pp. 797–813, Sep. 2018.

[35] W. J. Wang, S. Q. Yang, Q. Lin, Q. Zhang, K.-C. Wong, A. C. C. Coello, and J. Y. Chen, "An effective ensemble framework for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 645–659, Aug. 2019, doi: 10.1109/TEVC.2018.2879078.

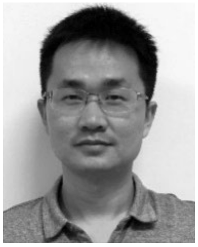
[36] Y. Qi, X. L. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[37] M. Tsiet, M. Gadala, "Self-adapting control parameters in particle swarm optimization," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105653.
[38] Y. Li, Y. Chen, J. Zhong, and Z. Huang, "Niching particle swarm optimization with equilibrium factor for multi-modal optimization," *Inf. Sci.*, vol. 494, pp. 233–246, Aug. 2019.
[39] J. Kennedy and R. Eberhart, "Particle swarm optimization (PSO)," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948.
[40] H. Han, W. Lu, L. Zhang, and J. Qiao, "Adaptive gradient multiobjective particle swarm optimization," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3067–3079, Nov. 2018.
[41] W. Peng and Q. Zhang, "A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems," in *Proc. IEEE Int. Conf. Granular Comput.*, Aug. 2008, pp. 534–537.
[42] X. Yu, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang, "Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2139–2153, Jul. 2018.
[43] N. AI Moubayed, A. Pertovski, and J. McCall, "D^2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 44–77, May 2014.
[44] R. Liu, J. Li, J. Fan, and L. Jiao, "A dynamic multiple populations particle swarm optimization algorithm based on decomposition and prediction," *Appl. Soft. Comput.*, vol. 73, pp. 434–459, Dec. 2018.
[45] A. A. Nagar, F. Han, Q.-H. Ling, and S. Mehta, "An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization," *IEEE Access*, vol. 7, pp. 50388–50399, 2019.
[46] H. Han, W. Lu, and J. Qiao, "An adaptive multiobjective particle swarm optimization based on multiple adaptive methods," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2754–2767, Sep. 2017.
[47] F. Li, J. Liu, S. Tan, and X. Yu, "R2-MOPSO: A multi-objective particle swarm optimizer based on R2-indicator and decomposition," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 3148–3155.
[48] L.-X. Wei, X. Li, R. Fan, H. Sun, and Z.-Y. Hu, "A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator," *IEEE Access*, vol. 6, pp. 14710–14721, 2018.
[49] X. Zhang, X. Wang, Q. Kang, and J. Cheng, "Differential mutation and novel social learning particle swarm optimization algorithm," *Inf. Sci.*, vol. 480, pp. 109–129, Apr. 2019.
[50] W. Hu, G. G. Yen, and G. Luo, "Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1446–1459, Jun. 2017.
[51] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019, doi: 10.1109/TEVC.2018.2875430.
[52] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[53] M. G. Gong, L. C. Jiao, H. F. Du, and L. F. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, pp. 225–255, Jun. 2008.
[54] Q. Z. Lin, Y. P. Ma, and J. Y. Chen, "An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies," *Inf. Sci.*, vols. 430–431, pp. 46–64, Mar. 2018.
[55] S. Q. Qian, Y. Q. Ye, and B. Jiang, "A micro-cloning dynamic multiobjective algorithm with adaptive change reaction strategy," *Soft. Comput.*, vol. 21, pp. 3781–3801, Jul. 2018.
[56] L. J. Li, Q. Z. Lin, S. B. Liu, D. W. Gong, C. A. C. Coello, and Z. Ming, "A novel multi-objective immune algorithm with a decomposition-based clonal selection," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105490.
[57] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

LINGJIE LI received the B.S. degree from the College of Computer Science and Technology, Shandong Technology and Business University, Yantai, China, in 2017. He is currently pursuing the M.S. degree with the School of Computer and Software, Shenzhen University. His current research interests include multiobjective optimization, Gaussian process-based evolutionary algorithm, immune clonal optimization, and particle swarm optimization.

SHUO CHEN received the B.S. degree from the College of Computer Science and Technology, Harbin University, Harbin, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer and Software, Shenzhen University. His current research interests include intrusion detection systems and multiobjective optimization.

ZHE GONG received the B.S. degree from Wuhan Polytechnic University, Wuhan, China, in 2015, and the M.S. degree from the South China University of Technology, Guangzhou, China, in 2018. He is currently a Research Assistant with the School of Computer and Software, Shenzhen University. His current research interests include discrete element method (DEM) and energy underground structure.



QIUZHEN LIN received the B.S. degree from Zhaoqing University, Zhaoqing, China, in 2007, the M.S. degree from Shenzhen University, Shenzhen, China, in 2010, and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2014.

He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University. He has published over ten research articles, since 2008. His current research interests include artificial immune systems, multiobjective optimization, and dynamic systems.

ZHONG MING is currently a Professor with the College of Computer and Software Engineering, Shenzhen University. He led two projects of the National Natural Science Foundation, including one key project (61836005) and one normal project (61672358). His major research interests include AI and cloud computing. He is a member of a council and a Senior Member of the China Computer Federation.