

## Research Article

# A Novel Angular-Guided Particle Swarm Optimizer for Many-Objective Optimization Problems

Fei Chen,<sup>1,2,3</sup> Shuhuan Wu,<sup>1,2,3</sup> Fang Liu ,<sup>1,2,3</sup> Junkai Ji ,<sup>1,2,3</sup> and Qiuzhen Lin <sup>1,2,3</sup>

<sup>1</sup>College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

<sup>2</sup>National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China

<sup>3</sup>Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China

Correspondence should be addressed to Fang Liu; [f.liu@szu.edu.cn](mailto:f.liu@szu.edu.cn)

Received 22 November 2019; Revised 4 March 2020; Accepted 12 March 2020; Published 17 April 2020

Academic Editor: Quanmin Zhu

Copyright © 2020 Fei Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most multiobjective particle swarm optimizers (MOPSOs) often face the challenges of keeping diversity and achieving convergence on tackling many-objective optimization problems (MaOPs), as they usually use the nondominated sorting method or decomposition-based method to select the local or best particles, which is not so effective in high-dimensional objective space. To better solve MaOPs, this paper presents a novel angular-guided particle swarm optimizer (called AGPSO). A novel velocity update strategy is designed in AGPSO, which aims to enhance the search intensity around the particles selected based on their angular distances. Using an external archive, the local best particles are selected from the surrounding particles with the best convergence, while the global best particles are chosen from the top 20% particles with the better convergence among the entire particle swarm. Moreover, an angular-guided archive update strategy is proposed in AGPSO, which maintains a consistent population with balanceable convergence and diversity. To evaluate the performance of AGPSO, the WFG and MaF test suites with 5 to 10 objectives are adopted. The experimental results indicate that AGPSO shows the superior performance over four current MOPSOs (SMPSO, dMOPSO, NMPSO, and MaPSO) and four competitive evolutionary algorithms (VaEA,  $\theta$ -DEA, MOEA/D-DD, and SPEA2-SDE), when solving most of the test problems used.

## 1. Introduction

Many real-world applications often face the problems of optimizing  $m$  (often conflicting) objectives [1]. This kind of engineering problem is called multiobjective optimization problems (MOPs), when  $m$  is 2 or 3, or called many-objective optimization problem (MaOPs), when  $m > 3$ . A generalized MOP or MaOP can be modeled as follows:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x)), \\ &\text{subject to } x \in \Omega, \end{aligned} \quad (1)$$

where  $x = (x_1, x_2, \dots, x_D)$  is a decision vector in the  $D$ -dimensional search space  $\Omega$  and  $F(x)$  defines  $m$  objective functions. Assuming that two solutions  $x$  and  $y$  ( $x \neq y$ ) locate in the search space  $\Omega$ , if  $f_i(x) \leq f_i(y)$ , for  $\forall i \in \{1, 2, \dots, m\}$ ,  $x$  is said to dominate  $y$ . Then, if no solution can dominate  $x$ , it is called nondominated solution.

For MOPs or MaOPs, the aim is to search a set of non-dominated solutions called Pareto-optimal set (PS), with its mapping in the objective space called Pareto-optimal Front (PF).

During the past two decades, a number of nature-inspired computational methodologies have been proposed to solve various kinds of MOPs and MaOPs, such as multiobjective evolutionary algorithms (MOEAs) [2, 3], multiobjective particle swarm optimizers (MOPSOs) [4–6], and multiobjective ant colony optimizers [7]. Especially, MOPSOs become one of the most outstanding population-based approaches due to the easy implementation and strong search ability [8], which can be also applied to other kinds of optimization problems, such as multimodal optimization problems [9] and standard image segmentation [10], as well as some real-world engineering problems [11, 12].

Currently, MOPSOs have been validated to be very effective and efficient when solving MOPs, which can search a set of approximate solutions with balanceable diversity and convergence to cover the entire PFs. Based on the selection of local or global best particles, which can effectively guide the flight of particles in the search space, most of MOPSOs can be classified into three main categories. The first category embeds the Pareto-based sorting approach [2] into MOPSO (called Pareto-based MOPSOs), such as OMOPSO [13], SMPSO [14], CMPSO [15], AMOPSO [5], and AGMOPSO [16]. The second category decomposes MOPs into a set of scalar subproblems and optimizes all the subproblems simultaneously using a collaborative search process (called decomposition-based MOPSOs), including dMOPSO [17], D<sup>2</sup>MOPSO [18], and MMOPSO [19]. The third category employs the performance indicators (e.g., HV [20] and R2 [21]) to guide the search process in MOPSOs (called indicator-based MOPSOs), such as IBPSO-LS [22], S-MOPSO [23], NMPSO [24], R2HMOPSO [25], and R2-MOPSO [26].

Although the above MOPSOs are effective for tackling MOPs with the objective number  $m \leq 3$ , their performance will significantly deteriorate for solving MaOPs with  $m > 3$ , mainly due to the several challenges brought by “the curse of dimensionality” in MaOPs [27, 28]. Generally, there are three main challenges for MOPSOs to deal with MaOPs.

The first challenge is to provide the sufficient selection pressure to approach the true PFs of MaOPs, i.e., the challenges in maintaining convergence. With the increase of objectives in MaOPs, it becomes very difficult for MOPSOs to pay the same attention to optimize all the objectives, which may lead to an imbalanced evolution such that solutions are very good at solving some objectives but perform poorly on the others [29]. Moreover, most solutions of MaOPs are often nondominated with each other at each generation. Therefore, MOPSOs based on Pareto dominance may lose the selection pressure on addressing MaOPs and show a poor convergence performance [24].

The second challenge is to search a set of solutions that are evenly distributed along the whole PF, i.e., the challenges in diversity maintenance. Since the objective space is enlarged rapidly with the increase of dimensionality in MaOPs, this requires a large number of solutions to approximate the whole PF. Some well-known diversity maintenance methods may not work well on MaOPs, e.g., the crowding distance [2] may prefer some dominated resistant solutions [30], the decomposition approaches require a larger set of well-distributed weight vectors [3], and the performance indicators require an extremely high computational cost [31]. Thus, diversity maintenance in MOPSOs becomes less effective with the increase of objectives in MaOPs [32].

The third challenge is to propose the effective velocity update strategies for MOPSOs to guide the particle search in the high-dimensional space [33]. There are a number of research studies to present the improved velocity update strategies for MOPs [34]. For example, in SMPSO [14], the velocity of the particles is constrained in order to avoid the cases in which the velocity becomes too high. In AgMOPSO [35], a novel archive-guided velocity update method is used to select the global best and personal best particles from the

external archive, which is more effective to guide the swarm. In MaOPSO/2s-pccs [36], a leader group is selected from the nondominated solutions with the better diversity in the external archive according to the parallel cell coordinates, which are selected as the global best solutions to balance the exploitation and exploration of a population. In NMPSO [24], another search direction from the personal best particle pointing to the global best one is provided to make more disturbances in velocity update. However, the abovementioned velocity update strategies are not so effective for MaOPs. With the enlargement of the objective space, it is more difficult to select the suitable personal or global best particles for velocity update.

To overcome the abovementioned challenges, this paper proposes a novel angular-guided MOPSO (called AGPSO). A novel angular-based archive update strategy is designed in AGPSO to well balance convergence and diversity in the external archive. Moreover, a density-based velocity update strategy is proposed to effectively guide the PSO-based search. When compared to the existing MOPSOs and some competitive MOEAs for MaOPs, the experimental results on the WFG and MaF test suites with 5 to 10 objectives have shown the superiority of AGPSO on most cases.

To summarize, the main contributions of this paper are clarified as follows.

- (1) An angular-guided archive update strategy is proposed in this paper. As the local-best and global-best particles are both selected from the external archive, the elitist solutions with balanceable convergence and diversity should be preserved to effectively guide the search direction to approximate the true PF. In this strategy,  $N$  evenly distributed particles ( $N$  is the swarm size) are firstly selected by the angular distances to maintain the diversity in different search directions, which helps to alleviate the second challenge above (diversity maintenance). After that, the rest particles are respectively associated to the closest particle based on the angular distance, which will form  $N$  groups with similar particles. At last, the particle with the best convergence in each particle group is saved in the external archive, trying to alleviate the first challenge above (keeping convergence).
- (2) A density-based velocity update strategy is designed in this paper, which can search high-quality particles with fast convergence and good distribution and helps to alleviate the third challenge above in MOPSOs. The local best and global best particles are selected according to the densities of the particles, for guiding the PSO-based search. The sparse particles will be guided by the local best particles to encourage exploitation in the local region as the surrounding particles are very few, while the crowded particles are disturbed by the global best particles to put more attentions on convergence. By this way, the proposed velocity update strategy is more suitable for solving MaOPs as experimentally validated in Section 4.5.

The rest of this paper is organized as follows. Section 2 introduces the background of the particle swarm optimizer

and some current MOPSOs and MOEAs for MaOPs. The details of AGPSO are given in Section 3. To validate the performance of AGPSO, some experimental studies are provided in Section 4. At last, our conclusions and future work are given in Section 5.

## 2. Background

**2.1. Particle Swarm Optimizer.** Particle swarm optimizer is a simple yet efficient swarm intelligent algorithm, which is inspired by the social behaviors of the individuals in flocks of birds and schools of fish, as proposed by Kennedy and Eberhart [6] in 1995. Generally, a particle swarm  $S$  is stochastically initialized to have  $N$  particles according to the corresponding search range of the problem ( $N$  is the swarm size). Each particle  $x_i \in S$  is a candidate solution to the problem, characterized with the velocity  $v_i(t)$  and the position  $x_i(t)$  at  $t$ th iteration.

Moreover, each particle  $x_i$  will record the historical position-information in  $S$ , i.e., the personal best position for  $x_i$  ( $pbest_i$ ) and the global best position for  $x_i$  ( $gbest_i$ ), which are used to guide the particle fly to the next position  $x_i(t+1)$ . Then, the velocity and location of each particle are iteratively updated as follows:

$$v_i(t+1) = wv_i(t) + c_1r_1(pbest_i - x_i(t)) + c_2r_2(gbest_i - x_i(t)), \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (3)$$

where  $w$  is an inertia weight to follow the previous velocity and  $r_1$  and  $r_2$  are two random numbers uniformly generated in  $[0, 1]$ , while  $c_1$  and  $c_2$  are the two learning weight factors from the personal and global best particles [14].

**2.2. Some Current MOPSOs and MOEAs for MaOPs.** Recently, a number of nature-inspired optimization algorithms have been proposed for solving MaOPs, such as MOEAs and MOPSOs. Especially, a number of MOEAs have been proposed, which can be classified into three main kinds, such as Pareto-based MOEAs, indicator-based MOEAs, and decomposition-based MOEAs. A lot of approaches have been proposed to enhance their performance for solving MaOPs. For example, Pareto dominance relation was modified to enhance the convergence pressure for Pareto-based MOEAs, such as  $\epsilon$ -dominance [30, 37]  $\theta$ -dominance, a strengthened dominance relation [38], and a special dominance relation [39]. New association mechanisms for solutions and weight vectors were designed for decomposition-based MOEAs, such as the use of reference vectors in NSGA-II [40], a dynamical decomposition in DDEA [41], and a self-organizing map-based weight vector design in MOEA/D-SOM [42]. A lot of effective and efficient performance indicators were presented for indicator-based MOEAs, such as a unary diversity indicator based on reference vectors [43], an efficient indicator as a combination of sum of objectives and shift-based density estimation called  $I_{SDE+}$  [44], and an enhanced inverted generational distance (IGD) indicator [45].

In contrast to the abovementioned MOEAs, there are only a few of MOPSOs presented to solve MaOPs. The reason may refer to the three challenges faced by MOPSOs as introduced in Section 1. The main difference between MOPSOs and MOEAs is their evolutionary search, e.g., most MOEAs adopt differential evolution or simulated binary crossover, while most MOPSOs use the fly of particles to run the search. The environmental selection in external archive of MOPSOs and MOEAs are actually similar. As inspired by the environmental selection in some MOEAs, MOPSOs can be easily extended to solve MaOPs. Recently, some competitive MOPSOs have been proposed to better solve MaOPs. For example, in pccsAMOPSO [5], a parallel cell coordinate system was employed to update the external archive for maintaining the diversity, which was used to select the global best and personal best particles. In its improved version MaOPSO/2s-pccs [36], the two-stage strategy was further presented to emphasize convergence and diversity, respectively, by using a single-objective optimizer and a many-objective optimizer. In NMPSO [24], a balanceable fitness estimation method was proposed by summarizing different weights of convergence and diversity factors, which aims to offer sufficient selection pressure in the search process. However, this method is very sensitive to the used parameters and requires a high-computational cost. In CPSO [29], a coevolutionary PSO with bottleneck objective learning strategy was designed for solving MaOPs. Multiple swarms coevolved in a distributed fashion to maintain diversity for approximating the entire PFs, while a novel bottleneck objective learning strategy was used to accelerate convergence for all objectives. In MaPSO [46], a novel MOPSO based on the acute angle was proposed, in which the leader of particles was selected from its historical particles by using the scalar projections and each particle owned  $K$  historical particle information ( $K$  was set to 3 in its experiments). Moreover, the environmental selection in MaPSO was run based on the acute angle of each pair of particles. Although these MOPSOs are effective for solving MaOPs, there are still some improvements for the design of MOPSOs. Thus, this paper proposes a novel angular-guided MOPSO with an angular-based archive update strategy and a density-based velocity update strategy to alleviate the three abovementioned challenges in Section 1.

## 3. The Proposed AGPSO Algorithm

In this section, the details of the proposed algorithm are provided. At first, two main strategies used in AGPSO, i.e., a density-based velocity update strategy and an angular-guided archive update strategy, are, respectively, introduced. The density-based velocity update strategy will consider the density around each particle, which is used to determine whether the local best particle or the global best particle is used to guide the swarm search. In this strategy, the local best particle is selected based on the local angular information, aiming to encourage exploitation in the local region around the current particle, while the global best particle is selected by the convergence performance, which is used to perform exploration in the whole search

space. In order to provide the elite particles for leading the PSO-based search, the angular-guided archive update strategy is designed to provide an angular search direction for each particle, which is more effective for solving MaOPs. At last, the complete framework of AGPSO is also shown in order to clarify its implementation as well as other components.

**3.1. A Density-Based Velocity Update Strategy.** As introduced in Section 2.1, the original velocity update strategy includes two best positional information of a particle, i.e., the personal-best (*pbest*) and global-best (*gbest*) particles, as defined in equation (1), which are used to guide the swarm search. Theoretically, the quality of generated particles is dependent on the guiding information from *pbest* and *gbest* in velocity update. However, the traditional velocity update strategy will face the great challenges in tackling MaOPs, as the selection of effective personal-best and global-best particles is difficult. Moreover, two velocity components used in equation (2) may lead to too much disturbance for the swarm search, as most of the particles are usually sparse in the high objective space. Thus, the quality of particles generated by the traditional velocity formula may not be promising, as the large disturbance in velocity update will lower the search intensity in the local region around the current particle. Some experiments have been given in Section 4.5 to study the effectiveness of different velocity update strategies. In this paper, a density-based velocity update strategy is proposed in equation (4), which can better control the search intensity in the local region of particles:

$$v_i(t+1) = \begin{cases} wv_i(t) + c_1r_1(lbest_i - x_i(t)), & D_x \geq \text{med}_{\text{SDE}}, \\ wv_i(t) + c_2r_2(gbest_i - x_i(t)), & D_x < \text{med}_{\text{SDE}}, \end{cases} \quad (4)$$

where  $t$  is the iteration number;  $v_i(t)$  and  $v_i(t+1)$  are the  $t$ th and the  $(t+1)$ th iteration velocity of particle  $x_i$ , respectively;  $w$  is the inertial weight;  $c_1$  and  $c_2$  are two learning factors;  $r_1$  and  $r_2$  are two uniformly distributed random numbers in  $[0, 1]$ ;  $lbest_i$  and  $gbest_i$  are the positional information of the local-best and global-best particles for  $x_i$ ,  $i = (1, 2, \dots, N)$ , respectively;  $D_x$  is the shift-based density estimation (SDE) of particle  $x$  as defined in [47]; and  $\text{med}_{\text{SDE}}$  is the median value of all  $D_x$  in the swarm. By this way, the sparse particles with  $D_x \geq \text{med}_{\text{SDE}}$  will be guided by the local best particles to encourage exploitation in the local region as their surrounding particles are very few, while the crowded particles with  $D_x < \text{med}_{\text{SDE}}$  are disturbed by the global best particles to put more attentions on convergence. Please note that  $lbest_i$  and  $gbest_i$  of  $x_i(t)$  are selected from the external archive  $A$  as introduced below.

To the selection of  $gbest_i$  of particle  $x_i$ , the selection of *gbest* focuses on the particles with better convergence in this archive. Meanwhile, we also need to ensure a certain perturbation; thus, we randomly choose from the top 20% of the convergence performance. The  $gbest_i$  of particle  $x_i$  is randomly selected from the top 20% particles in the external archive with the better convergence performance values. For

each particle  $x$ , we calculate its convergence performance values using the following formulation:

$$\text{Con}(x) = \sum_{k=1}^m f'_k(x), \quad (5)$$

where  $m$  is the number of objectives and  $f'_k(x)$  returns the  $k$ th normalized objective vector of  $x$ , which is obtained by the normalization procedure as follows:

$$f'_k(x) = \frac{f_k(x) - z_k^*}{z_k^{\text{nadir}} - z_k^*}, \quad (6)$$

where  $z_k^*$  and  $z_k^{\text{nadir}}$  are the  $k$ th objective value of the ideal and nadir points, respectively,  $k = 1, 2, \dots, m$ .

To the selection of  $lbest_i$ , the current particle  $x_i$  in the particle swarm ( $i = 1, \dots, N$ , where  $N$  is the number of particles size) will be firstly associated to the closest particle  $y$  in the external archive ( $A$ ), by comparing the angular-distance of  $x_i$  to each particle in the extra archive ( $A$ ). And use  $y$  particle to find the  $lbest_i$  in  $A$ . Here, the angular-distance [41] between the particle  $x$  to another particle  $y$ , termed  $AD(x, y)$ , is computed by

$$AD(x, y) = \|\lambda(x) - \lambda(y)\|_2, \quad (7)$$

where  $\lambda(x)$  is defined as follows:

$$\lambda(x) = \frac{1}{\sum_{k=1}^m f'_k(x)} F_I(x), \quad (8)$$

where  $F_I(x) = (f'_1(x), f'_2(x), \dots, f'_m(x))^T$  and  $f_I(x)$  is calculated by equation (6).

Then,  $lbest_i$  is selected from the  $T$  angular-distance-based closest neighbors of  $y$  with the best convergence value by equation (5). Specially, each particle of  $A$  owns their  $T$  nearest neighbors, which are found by calculating the angular distance of each particle to the other particles of  $A$  using equation (7). The neighborhood information of  $x_i$  in  $A$  is obtained in  $B(i)$ , where  $B(i) = \{i_1, i_2, \dots, i_T\}$  and  $(x_{i_1}, x_{i_2}, \dots, x_{i_T})$  are  $T$  closest angular-distance-based solutions to  $x_i$  by equation (7).

Then, for each particle in  $S$ ,  $lbest_i$  and  $gbest_i$  are obtained to update the velocity by equation (4) and position by equation (3), respectively. After that, the object values of the particle are evaluated. To further clarify the selection of local best and global best particles and the density-based velocity update, the related pseudocode is given in Algorithm 1.

**3.2. An Angular-Guided Archive Update Strategy.** After performing the abovementioned density-based velocity update for the swarm search, each particle flies to the new position to produce the new particle swarm  $S$ . In order to get a population consisting of a fixed number of elitist solutions, which can maintain an excellent balance between convergence and distribution in the external archive, an appropriate selection strategy is required to update  $A$ . In this paper, we propose an angular-guided archive update strategy, following the principle of distributivity first and convergence second, and the pseudocode of updating the external archive is given in Algorithm 2.

```

(1) sort the extra archive  $A$  in ascending order based on convergence value as in equation (5)
(2) for each solution  $x_i \in A, i = (1, 2, \dots, N)$ 
(3)   identify its  $T$  neighborhood index as  $B(i)$ 
(4) end for
(5) for each particle  $x_i \in S, i = (1, 2, \dots, N)$ 
(6)   associate  $x_i$  to the angular-guided closest particle  $y \in A$  with index  $j$  in  $A$ 
(7)   get the  $T$  nearest neighbors in  $A$  of  $y$  by the neighborhood index  $B(j)$ 
(8)   sort the  $T$  neighbors of  $y$  in an ascending order based on convergence value by equation (5)
(9)   select the first angle-distance-based neighboring particle of  $y$  as lbesti
(10)  select randomly from the top 20% particles in  $A$  as gbesti
(11)  update the velocity  $v_i$  of  $x_i$  by equation (4)
(12)  update the position  $x_i$  by equation (3)
(13)  evaluate the objective values for  $x_i$ 
(14) end for
(15) return  $S$ 

```

ALGORITHM 1: Density-based velocity update ( $S, A, \text{med}_{\text{SDE}}$ ).

In Algorithm 2, the input is  $S$  and  $A$ ;  $N$  is the size of both  $S$  and  $A$ . First,  $S$  and  $A$  are combined into a union set  $U$  and set  $A = \emptyset$  in line 1. Then, all the particles in  $U$  is normalized by equation (6) in line 2. After that, in lines 3–6, a set of particles are found to represent the distribution of all particles. To do this, each time, the current particle pair  $(x_h, x_u)$  with minimum angular-distance of  $U$  is found in line 4, which is computed as

$$(x_h, x_u) = \operatorname{argmin}_{(x_h, x_u)} \{AD(x_h, x_u) | x_h, x_u \in U\}, \quad (9)$$

where the  $AD(x, y)$  is computed as equation (7). Then, one particle  $x^*$  in  $(x_h, x_u)$  with the minimum angular-distance to  $U$  is found and added to  $A$  and deleted from  $U$  in lines 5–6. Here, the angular-distance of  $(x_h, x_u)$  to  $U$ , termed as  $AD(x, U)$ , is computed as

$$AD(x, U) = \operatorname{minimize} \{AD(x, y) | y \in U, y \notin (x_h, x_u), x \in (x_h, x_u)\}, \quad (10)$$

where the  $AD(x, y)$  is computed as equation (7),  $N$  subsets  $S_1, S_2, \dots, S_N$  are obtained in line 8–11, where particle  $x_i$  in  $U$  is saved into  $S_i, i = 1, 2, \dots, N$ . In lines 12–15, each particle  $x$  in  $A$  is associated with the minimum angular-distance to the particle  $x_i$  in the  $U$ , and then this particle  $x$  of  $A$  is added into the subset  $S_i, i = 1, 2, \dots, N$  in line 14. Then, set  $A = \emptyset$  in line 16. Finally, in each subset  $S_i, i = 1, 2, \dots, N$ , a particle with the best convergence performance computed by equation (5) is added into  $A$ , in lines 17–20. The  $A$  is returned in line 21 as the final result.

In order to facilitate the understanding of the process of this angular-guided strategy, a simple example is illustrated in Figure 1. The  $U$  includes ten individuals  $x_1, x_2, \dots, x_{10}$  and their mapping solutions, which map individuals to the hyperplane  $f'_1 + f'_2 = 1$  (calculating angular-distance) in the normalized biobjective space, as shown in Figure 1(a). First, five particles (half the population size), i.e.,  $x_1, x_4, x_7, x_9, x_{10}$  that represent the distribution of all ten particles are kept in  $U$ , as shown in Figure 1(b). Then, the remaining particles, i.e.,  $x_2, x_3, x_5, x_6, x_8$ , are preserved in  $A$  as in Figure 1(c). Second, each particle in Figure 1(c) is

associated with a minimum angular-distance particle in Figure 1(b). After that, five subsets  $S_1, S_2, \dots, S_5$  are obtained, where  $S_1$  preserves  $x_1$  and its associated particles in  $A$ , i.e.,  $x_2$ . Similarly,  $S_2$  preserves  $x_3$  and  $x_4$ ;  $S_3$  preserves  $x_5, x_6$ , and  $x_7$ ;  $S_4$  preserves  $x_8$  and  $x_9$ ;  $S_5$  preserves  $x_{10}$ , as shown in Figure 1(d). Finally, in each subset, only the particle with the best convergence is selected, as in Figure 1(e).

**3.3. The Complete Algorithm of AGPSO.** The above sections have introduced the main components of AGPSO, which include the velocity update strategy and archive update operator. In addition, the evolutionary search is also proposed on the external archive. In order to describe the remaining operator and to facilitate the implementation of AGPSO, the pseudocode of its complete algorithm is provided in Algorithm 3. The initialization procedure is first activated in lines 1–6 of Algorithm 3. For each particle in  $S$ , its positional information is randomly generated and its velocity is set to 0 in line 3. After that the objectives of each particle are evaluated in line 4. The external archive  $A$  is updated with Algorithm 2 in line 7 and sorted in the ascending order by calculating the convergence fitness of each particle using equation (5) in line 8. After that the iteration counter  $t$  is increased by one. Then, AGPSO steps into the main loop, which contains the particle search and the evolutionary search, until the maximum number of iterations is reached.

In the main loop, the SDE distance of each particle in  $S$  is calculated, and these particles are sorted by the SDE distance to get median SDE distance of particles in  $S$ , which is called  $\text{med}_{\text{SDE}}$  as in lines 11–12. Then, the PSO-based search is performed in line 13 with density-based velocity update of Algorithm 1. After that the angular-guided archive update strategy of Algorithm 2 is executed in line 14, with the inputs  $S, A$ , and  $N$  (the sizes of  $S$  and  $A$  are both  $N$ ). Then, in line 15, the evolutionary strategy is applied on  $A$  to optimize the swarm leaders, providing another search strategy to cooperate with the PSO-based search.

```

(1) combine  $S$  and  $A$  into a union set  $U$  and set  $A = \emptyset$ 
(2) normalize all particles in  $U$  by equation (6)
(3) for  $i = 1$  to  $N$ 
(4)   find the particle pair  $(x_l, x_u)$  in  $U$  with the minimum angular-distance for all particle pairs in  $U$ 
(5)   find  $x^*$  in  $(x_l, x_u)$  such that  $x^*$  has the smaller angular-distance to  $U$  by equation (9)
(6)   add  $x^*$  to  $A$ , then deleted  $x^*$  from  $U$ 
(7) end for
(8) for each particle  $x_i \in U$ 
(9)   initialize an empty subset  $S_i$ 
(10)  add the particle  $x_i$  into  $S_i$ 
(11) end for
(12) for each particle  $x_i \in A$ 
(13)  associate  $x_i$  to the particle  $x_t$  in  $U$  that has the smallest angular-distance to  $x_i$ 
(14)  add  $x_i$  into  $S_t$ 
(15) end for
(16) set  $A = \emptyset$ 
(17) for  $i = 1$  to  $N$ 
(18)  find the particle  $x^*$  of  $S_i$  that has the best convergence performance computed by equation (5)
(19)  add  $x^*$  into  $A$ 
(20) end for
(21) return  $A$ 

```

ALGORITHM 2: Angular-guided archive update ( $S, A, N$ ).

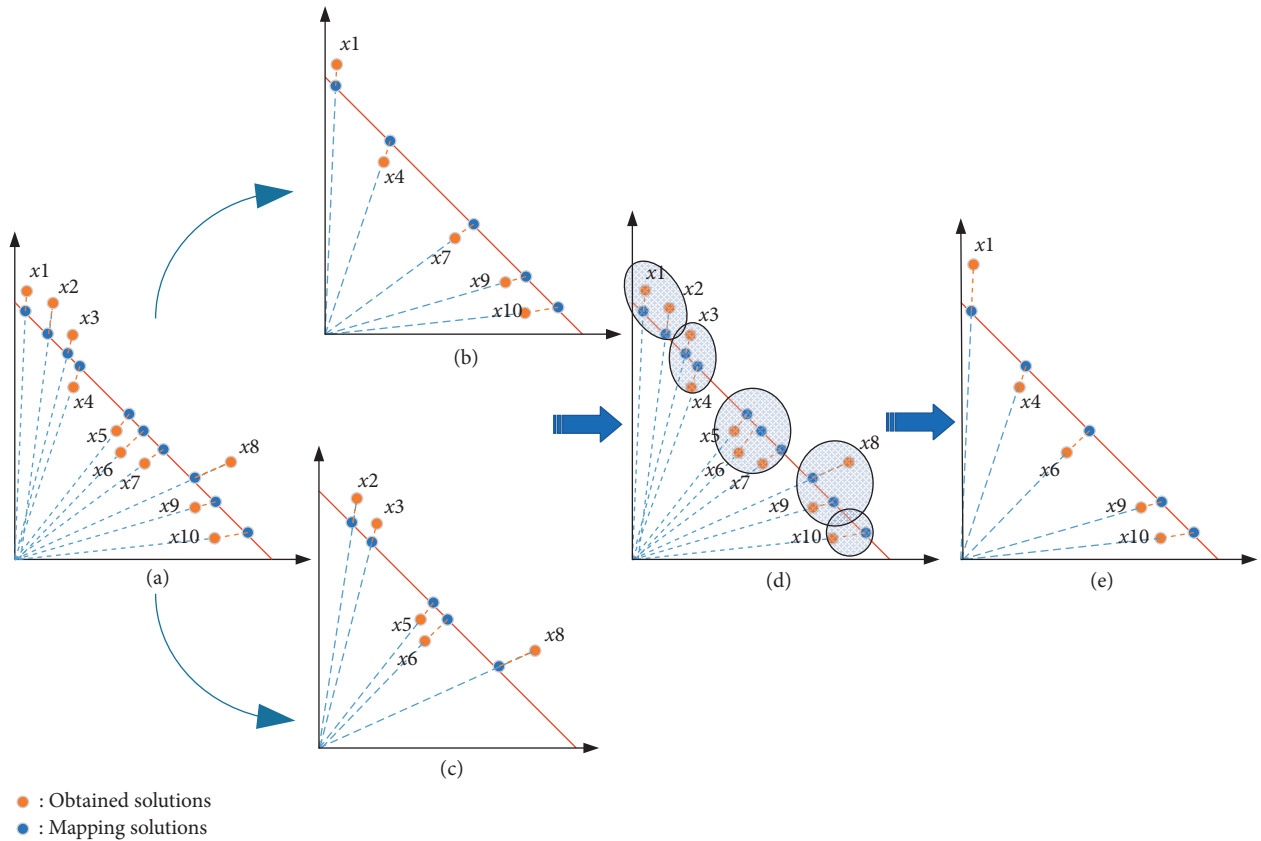


FIGURE 1: An example to illustrate the process of the proposed angular-guided archive update strategy.

Finally, the objectives of solutions, which are newly generated by the evolutionary strategy, are evaluated in line 16. Also, the angular-guided archive update strategy of

Algorithm 2 is carried out again in line 17, and the  $A$  is sorted in the ascending order again in line 18 for selecting the global-best (gbest) particles. The iteration counter  $t$  is

```

(1) Let  $t=0$ ,  $A = \emptyset$ , initial particles  $S = \{x_1, x_2, \dots, x_N\}$ , and  $T$  be the value of the neighborhood size
(2) for  $i = 1$  to  $N$ 
(3)   randomly initialize position  $x_i$  and set  $v_i = 0$  for  $x_i$ 
(4)   evaluate the objective values of  $x_i$ 
(5) end for
(6) randomly initialize  $A$  with  $N$  new particles
(7)  $A = \mathbf{Angular-guided\ Archive\ update}(S, A, N)$ 
(8) sort  $A$  in ascending order based on equation (5)
(9)  $t = t + 1$ 
(10) while  $t < t_{\max}$ 
(11)   calculate the SDE of each particle in  $S$ 
(12)   sort particles in  $S$  to get the  $\text{med}_{\text{SDE}}$ 
(13)    $S^{\text{new}} = \mathbf{Density-based\ Velocity\ Update}(S, A, \text{med}_{\text{SDE}})$ 
(14)    $A = \mathbf{Angular-guided\ Archive\ update}(S^{\text{new}}, A, N)$ 
(15)   apply evolutionary search strategy on  $A$  to get a new swarm  $S^{\text{new}}$ 
(16)   evaluate the objectives of the new particles in  $S^{\text{new}}$ 
(17)    $A = \mathbf{Angular-guided\ Archive\ update}(S^{\text{new}}, A, N)$ 
(18)   sort  $A$  in ascending order based on equation (5)
(19)    $t = t + 2$ 
(20) end while
(21) output  $A$ 

```

ALGORITHM 3: The complete algorithm of AGPSO.

increased by 2 in line 19 because one PSO-based search and one evolutionary strategy are carried out in each particle and in each of the main loop. The abovementioned operation is repeated until the maximum of iterations ( $t_{\max}$ ) is achieved. At last, the final particles in  $A$  are saved as the final approximation of PF.

## 4. The Experimental Studies

### 4.1. Related Experimental Information

**4.1.1. Involved MOEAs.** Four competitive MOEAs are used to evaluate the performance of our proposed AGPSO. They are briefly introduced as follows.

- (1) VaEA [48]: this algorithm uses the maximum-vector-angle-first principle in the environmental selection to guarantee the wideness and uniformity of the solution set.
- (2)  $\theta$  - DEA [30]: this algorithm uses a new dominance relation to enhance the convergence in the high dimension optimization.
- (3) MOEA/DD [49]: this algorithm proposes to exploit the merits of both dominance-based and decomposition-based approaches, which balances the convergence and the diversity for the population.
- (4) SPEA2 + SDE [47]: this algorithm develops a general modification of density estimation in order to make Pareto-based algorithms suitable for many-objective optimization.

Four competitive MOPSOs are also used to validate the performance of our proposed AGPSO. They are also briefly introduced as follows:

- (1) NMPSO [24]: this algorithm uses a balanceable fitness estimation to offer sufficient selection pressure in the search process, which considers both of the convergence and diversity with weight vectors.
- (2) MaPSO [46]: this algorithm uses an angle-based MOPSO called MaPSO. MaPSO chooses the leader positional information of particles from its historical particles by using scalar projections to guide particles.
- (3) dMOPSO [17]: this algorithm integrates the decomposition method to translate an MOP into a set of single-objective optimization problems, which solves them simultaneously using a collaborative search process by applying PSO directly.
- (4) SMP SO [14]: this algorithm proposes a strategy to limit the velocity of the particles.

**4.1.2. Benchmark Problems.** In our experiments, sixteen various unconstrained MOPs are considered here to assess the performance of the proposed AGPSO algorithm. Their features are briefly introduced below.

In this study, the WFG [50] and MaF [31] test problems were used, including WFG1-WFG9 and MaF1-MaF7. For each problem, the number of objectives was varied from 5 to 10, i.e.,  $m \in \{5, 8, 10\}$ . For MaF1-MaF7, the number of decision variables was set as  $n = m + k - 1$ , where  $n$  and  $m$  are, respectively, the number of decision variables and the number of objectives. As suggested in [31], the values of  $k$  were set to 10. Regarding WFG1-WFG9, the decision variables are composed by  $k$  position-related parameters and  $l$  distance-related parameters. As recommended in [51],  $k$  is set to  $2 \times (m - l)$  and  $l$  is set to 20.

**4.1.3. Performance Metrics.** The goals of MaOPs include the minimization of the distance of the solutions to the true PF (i.e., convergence) and the maximization of the uniform and spread distribution of solutions over the true PF (i.e., diversity).

Hypervolume (HV) [20] metric calculates the objective space between the prespecified reference point  $z^r = (z_1^r, \dots, z_m^r)^T$  which is dominated by all points on the true points and the solution set  $S$  we obtain. The calculation of HV metric is given as follows:

$$\text{HV}(S) = \text{Vol}\left(\bigcup_{x \in S} [f_1(x), z_1^r] \times \dots \times [f_m(x), z_m^r]\right), \quad (11)$$

where  $\text{Vol}(\cdot)$  denotes the Lebesgue measure. When calculating the HV value, the points are firstly normalized as suggested in [51] by the vector  $1.1 \times (f_1^{\max}(x), f_2^{\max}(x), \dots, f_m^{\max}(x))$  with  $f_k^{\max}(x)$  ( $k = 1, 2, \dots, m$ ) being the maximum value of  $k$ th objective in true PF. Those that cannot dominate the reference point will be discarded (i.e., solutions that cannot dominate the reference point are not included to compute HV). For each objective, an integer larger than the worst value of the corresponding objective in the true PF is adopted as the reference point. After the normalization operation, the referent point is set to  $(1.0, 1.0, \dots, 1.0)$ . A larger value of HV indicates a better approximation of the true PF.

**4.2. General Experimental Parameter Settings.** In this paper, four PSO algorithms (dMOPSO [17], SMPSO [14], NMPSO [24], and MaPSO [46]), and four competitive MOEAs (VaEA,  $\theta$ -DEA, MOEA/DD, and SPEA2 + SDE), were used for performance comparison.

Because the weight vectors are implied to dMOPSO and MOEA/DD, the population sizes of these two algorithms are set the same as the number of weight vectors. The number of weight vectors is set to 210, 240, and 275, following [51], for the test problem with 5, 8, and 10 objectives. In order to ensure fairness, the other algorithms adopt the population/swarm size the same as to the number of weight vectors.

To allow a fair comparison, the related parameters of all the compared algorithms were set as suggested in their references, as summarized in Table 1.  $p_c$  and  $p_m$  are the crossover probability and mutation probability;  $\eta_c$  and  $\eta_m$  are, respectively, the distribution indexes of SBX and polynomial-based mutation. For these PSO-based algorithms mentioned above (SMPSO, dMOPSO, NMPSO, AGPSO), the control parameters  $c_1, c_2, c_3$  are sampled in  $[1.5, 2.5]$ , and the inertia weight  $w$  is a random number of  $[0.1, 0.5]$ . In MaPSO,  $K$  is the size of historical maintained by each particle, which is set to 3; the another algorithmic parameters are  $\theta_{\max} = 0.5$ ,  $w = 0.1$ , and  $C = 2.0$ . In MOEA/DD,  $T$  is the neighborhood size;  $\delta$  is the probability to select parent solutions from  $T$  neighborhoods; and  $n_r$  is the maximum number of parent solutions that are replaced by each child solution. Regarding VaEA,  $\sigma$  is a condition for determining whether the solution is searching for a similar direction, which is set to  $\pi/2(N + 1)$ .

TABLE 1: General experimental parameter settings.

Algorithm	Parameter Settings
SMPSO	$w \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
dMOPSO	$w \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5]$
NMPSO	$w \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
MaPSO	$K = 3, \theta_{\max} = 0.5, C = 2.0, w = 0.1$
VaEA	$p_c = 1.0, p_m = 1/n, \eta_m = 20, \sigma = \pi/2(N + 1)$
$\theta$ -DEA	$p_c = 1.0, p_m = 1/n, \eta_m = 20, \eta_c = 30$
MOEA/DD	$p_c = 1.0, p_m = 1/n, \eta_m = 20, \eta_c = 30, T = 20, \delta = 0.9$
SPEA2 + SDE	$p_c = 1.0, p_m = 1/n, \eta_m = 20, \eta_c = 30$

All the algorithms were run 30 times independently on each test problem. The mean HV values and the standard deviations (included in bracket after the mean HV results) in 30 runs were collected for comparison. All the algorithms were terminated when a predefined maximum number of generations ( $t_{\max}$ ) was reached. In this paper,  $t_{\max}$  is set to 600, 700, and 1000 for 5-, 8-, and 10-objective problems, respectively. For each algorithm, the maximum function evaluations (MFE) can be easily determined by  $\text{MFE} = N * t_{\max}$ , where  $N$  is the population size. To obtain a statistically sound conclusion, Wilcoxon rank sum test was run with a significance level (0.05 is set in this paper), showing the statistically significant differences on the results of AGPSO and other competitors. In the following experimental results, the symbols “+,” “-,” and “~” indicate that the results of other competitors are significantly better than, worse than, and similar to the ones of AGPSO using this statistical test, respectively.

All these nine algorithms were implemented by JAVA codes and run on a personal computer with Intel (R) Core (TM) i7- 6700 CPU, 3.40 GHz (processor), and 20 GB (RAM).

**4.3. Comparison with State-of-the-Art Algorithms.** In this section, AGPSO is compared to four PSO algorithms (dMOPSO, SMPSO, NMPSO, and MaPSO) and four competitive MOEAs (VaEA,  $\theta$ -DEA, MOEA/DD, and SPEA2 + SDE) on WFG1-WFG9 and MaF1-MaF7 problems. 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 AGPSO using statistical test, respectively. The best mean result for each problem is highlighted in boldface.

#### 4.3.1. Comparison Results with Four Current MOPSOs

(1) *Comparison Results on WFG1-WFG9.* Table 2 shows the HV performance comparisons of four PSOs on WFG1-WFG9, which clearly demonstrate that AGPSO provides promising performance in solving WFG problems with these PSOs, as it is best on 21 out of 27 cases on the experimental data. In contrast, NMPSO, MaPSO, dMOPSO, and SMPSO are best on 6, 1, 0, and 0 cases for HV metric, respectively. These data are summarized in the second last row of Table 2. The last row of Table 2 are the comparisons of AGPSO with each PSO on WFG, where the mean of “-/~/+” is the number of test problem in which the corresponding



TABLE 2: Comparison of results of AGPSO and four current MOPSOs on WFG1-WFG9 using HV.

Problem	Obj	AGPSO	NMPSO	MaPSO	dMOPSO	SMPSO
WFG1	5	<b>7.87e-01(4.27e-02)</b>	5.97e-01(2.89e-02)-	3.34e-01(2.58e-02)-	2.98e-01(3.44e-03)-	3.00e-01(1.01e-03)-
	8	<b>8.75e-01(3.77e-02)</b>	7.00e-01(1.38e-01)-	2.74e-01(2.45e-02)-	2.35e-01(6.26e-03)-	2.50e-01(1.54e-03)-
	10	<b>9.45e-01(3.19e-03)</b>	8.07e-01(1.58e-01)-	2.53e-01(1.98e-02)-	2.13e-01(9.18e-03)-	2.30e-01(1.29e-03)-
WFG2	5	<b>9.76e-01(5.76e-03)</b>	9.69e-01(5.44e-03)-	9.70e-01(4.70e-03)-	8.94e-01(1.28e-02)-	8.85e-01(1.31e-02)-
	8	9.82e-01(4.12e-03)	9.83e-01(3.56e-03)+	<b>9.84e-01(2.86e-03)+</b>	8.92e-01(1.48e-02)-	8.48e-01(2.52e-02)-
	10	<b>9.93e-01(1.99e-03)</b>	9.91e-01(4.85e-03)~	9.90e-01(1.48e-03)-	9.00e-01(1.76e-02)-	8.54e-01(2.55e-02)-
WFG3	5	<b>6.59e-01(5.16e-03)</b>	6.13e-01(1.94e-02)-	6.12e-01(1.64e-02)-	5.89e-01(1.05e-02)-	5.75e-01(7.32e-03)-
	8	<b>6.78e-01(6.20e-03)</b>	5.77e-01(1.52e-02)-	5.96e-01(1.88e-02)-	4.94e-01(4.58e-02)-	5.84e-01(1.51e-02)-
	10	<b>6.91e-01(7.88e-03)</b>	5.83e-01(2.73e-02)-	5.97e-01(2.58e-02)-	3.56e-01(9.58e-02)-	5.92e-01(1.52e-02)-
vWFG4	5	<b>7.74e-01(5.34e-03)</b>	7.64e-01(6.09e-03)-	7.22e-01(2.96e-02)-	6.51e-01(1.08e-02)-	5.09e-01(1.73e-02)-
	8	<b>9.00e-01(8.05e-03)</b>	8.62e-01(1.21e-02)-	7.98e-01(1.76e-02)-	3.88e-01(7.86e-02)-	5.26e-01(2.13e-02)-
	10	<b>9.37e-01(6.64e-03)</b>	8.82e-01(8.54e-03)-	8.27e-01(1.32e-02)-	3.15e-01(1.34e-01)-	5.52e-01(2.19e-02)-
WFG5	5	<b>7.41e-01(6.12e-03)</b>	7.35e-01(3.73e-03)-	6.63e-01(1.18e-02)-	5.85e-01(1.53e-02)-	4.30e-01(1.18e-02)-
	8	<b>8.61e-01(4.35e-03)</b>	8.32e-01(6.58e-03)-	6.91e-01(1.12e-02)-	1.81e-01(8.43e-02)-	4.52e-01(1.08e-02)-
	10	<b>8.88e-01(8.29e-03)</b>	8.65e-01(7.92e-03)-	6.99e-01(1.76e-02)-	2.43e-02(1.54e-02)-	4.65e-01(1.24e-02)-
WFG6	5	<b>7.50e-01(8.96e-03)</b>	7.24e-01(3.96e-03)-	7.09e-01(4.50e-03)-	6.74e-01(7.93e-03)-	5.43e-01(1.66e-02)-
	8	<b>8.72e-01(1.26e-02)</b>	8.08e-01(5.70e-03)-	8.03e-01(2.66e-03)-	4.89e-01(4.87e-02)-	6.49e-01(8.38e-03)-
	10	<b>9.09e-01(1.36e-02)</b>	8.34e-01(2.03e-03)-	8.32e-01(1.80e-03)-	4.66e-01(2.96e-02)-	7.03e-01(1.37e-02)-
WFG7	5	7.90e-01(2.12e-03)	<b>7.96e-01(2.98e-03)+</b>	7.89e-01(4.61e-03)~	5.39e-01(2.72e-02)-	4.44e-01(2.06e-02)-
	8	<b>9.20e-01(3.39e-03)</b>	9.04e-01(3.71e-03)-	8.73e-01(1.40e-02)-	2.25e-01(3.93e-02)-	4.73e-01(1.23e-02)-
	10	<b>9.55e-01(9.11e-03)</b>	9.36e-01(6.97e-03)-	9.17e-01(1.17e-02)-	2.07e-01(6.36e-02)-	5.14e-01(2.10e-02)-
WFG8	5	6.63e-01(3.75e-03)	<b>6.73e-01(5.41e-03)+</b>	6.29e-01(1.34e-02)-	4.27e-01(1.57e-02)-	3.72e-01(2.09e-02)-
	8	7.85e-01(7.46e-03)	<b>7.97e-01(3.79e-02)~</b>	6.91e-01(2.13e-02)-	9.55e-02(3.20e-02)-	4.11e-01(9.29e-03)-
	10	8.36e-01(1.30e-02)	<b>8.62e-01(4.12e-02)+</b>	7.33e-01(2.04e-02)-	7.73e-02(1.27e-02)-	4.52e-01(1.90e-02)-
WFG9	5	6.57e-01(3.82e-03)	<b>7.35e-01(7.32e-03)+</b>	6.23e-01(7.01e-03)-	5.97e-01(1.99e-02)-	4.46e-01(1.61e-02)-
	8	7.30e-01(9.96e-03)	<b>7.65e-01(6.80e-02)+</b>	6.55e-01(1.17e-02)-	2.73e-01(6.06e-02)-	4.55e-01(2.39e-02)-
	10	<b>7.49e-01(9.03e-03)</b>	7.47e-01(6.72e-02)-	6.56e-01(1.30e-02)-	2.24e-01(6.96e-02)-	4.76e-01(1.23e-02)-
Best/all		20/27	6/27	1/27	0/27	0/27
Worse/ similar/better			19-/2~/6+	25-/1~/1+	27-/0~/0+	27-/0~/0+

algorithm performs worse than, similarly than, better than the AGPSO.

It is found that the AGPSO shows absolute advantage from the comparisons with NMPSO, MaPSO, dMOPSO, and SMPSO on the five test problems: WFG1 with a convex and mixed PF; WFG3 with a linear and degenerate PF; WFG4-WFG6 with concave PF. Regarding WFG2, which has a disconnected and mixed PF, AGPSO shows a better performance than the other PSOs except MaPSO. AGPSO is the best on WFG2 with 5 and 8 objectives, while the MaPSO is best on WFG2 with 10 objectives. For WFG7 with concave PF, nonseparability of position, and distance parameters, AGPSO is worse than NMPSO with 5 objectives, but better with 8 and 10 objectives. Regarding WFG8, where distance related parameters are dependent on position related parameters, AGPSO performs worse with 5 and 10 objectives and similar with 8 objectives. For WFG9 with a multimodal and deceptive PF, AGPSO only performs the best with 10 objectives and worse with 5 and 8 objectives.

As observed from the one-to-one comparisons in the last row of Table 2, SMPSO and dMOPSO perform poorly in solving WFG problems; AGPSO showed the superior performance over the traditional PSOs on WFG problems.

From the results, AGPSO is better than NMPSO and MaPSO in 19 and 25 out of 27 cases, respectively. Conversely, it was worse than NMPSO and MaPSO in 6 and 1 out of 27 comparisons, respectively. Therefore, it is reasonable to conclude that AGPSO showed a superior performance over NMPSO and MaPSO in most problems of WFG1-WFG9.

(2) *Comparison Results on MaF1-MaF7.* Table 3 provides the mean HV comparison results of AGPSO with four PSO algorithms (NMPSO, MaPSO, dMOPSO, and SMPSO) on MaF1-MaF7 with 5, 8, and 10 objectives. As observed from the second last row in Table 3, there are 12 best results in 21 test problems obtained by AGPSO, while NMPSO performs best in 7 cases, MaPSO and SMPSO perform best only in 1 case, respectively, and dMOPSO is not best on any MaF test problem.

MaF1 is modified inverted DTLZ1 [52], which leads to the shape of reference points not able to fit to the PF shape of MAFL. The performance of reference point-based PSO (dMOPSO) is worse than PSO algorithms that do not use reference points (AGPSO, NMPSO, and MaPSO), while the AGPSO performs best on tackling the MaF1 test problem. MaF2 is obtained from DTLZ2 to increase the difficulty of convergence, which requires that all objectives are optimized

TABLE 3: Comparison of results of NMPSO and four current MOPSOs on MAF1-MAF7 using HV.

Problem	Obj	AGPSO	NMPSO	MaPSO	dMOPSO	SMPSO
MaF1	5	<b>1.31e-02(1.43e-04)</b>	1.27e-02(5.81e-05)-	1.15e-02(2.80e-04)-	1.06e-02(1.26e-04)-	6.83e-03(5.14e-04)-
	8	<b>4.23e-05(1.80e-06)</b>	3.64e-05(7.99e-07)-	2.79e-05(2.66e-06)-	1.30e-05(1.49e-05)-	1.21e-05(1.88e-06)-
	10	<b>6.16e-07(3.49e-08)</b>	4.75e-07(1.83e-08)-	3.83e-07(2.83e-08)-	8.82e-08(8.92e-08)-	1.46e-07(2.86e-08)-
MaF2	5	<b>2.64e-01(1.18e-03)</b>	2.63e-01(8.71e-04)-	2.38e-01(3.57e-03)-	2.50e-01(1.55e-03)-	2.12e-01(4.05e-03)-
	8	<b>2.46e-01(2.64e-03)</b>	2.36e-01(2.32e-03)-	2.09e-01(3.04e-03)-	2.02e-01(2.13e-03)-	1.74e-01(4.46e-03)-
	10	<b>2.45e-01(2.08e-03)</b>	2.23e-01(2.81e-03)-	2.06e-01(5.67e-03)-	2.05e-01(2.15e-03)-	1.88e-01(4.28e-03)-
MaF3	5	<b>9.88e-01(2.87e-03)</b>	9.71e-01(1.47e-02)-	9.26e-01(5.29e-02)-	9.38e-01(8.27e-03)-	7.34e-01(4.50e-01)-
	8	9.95e-01(2.11e-03)	<b>9.99e-01(7.50e-04)+</b>	9.84e-01(1.48e-02)-	9.62e-01(1.44e-02)-	0.00e+00(0.00e+00)-
	10	9.85e-01(2.26e-03)	<b>1.00e+00(1.92e-04)+</b>	9.95e-01(6.00e-03)~	9.71e-01(9.43e-03)-	0.00e+00(0.00e+00)-
MaF4	5	<b>1.29e-01(1.22e-03)</b>	5.43e-02(3.25e-02)-	1.01e-01(1.07e-02)-	4.21e-02(4.98e-03)-	7.96e-02(3.93e-03)-
	8	<b>6.02e-03(9.67e-05)</b>	2.29e-03(7.77e-04)-	3.18e-03(6.84e-04)-	1.24e-04(7.69e-05)-	1.82e-03(3.39e-04)-
	10	<b>5.66e-04(1.89e-05)</b>	2.10e-04(1.26e-04)-	2.36e-04(6.20e-05)-	1.55e-06(8.26e-07)-	1.12e-04(2.32e-05)-
MaF5	5	8.00e-01(2.70e-03)	<b>8.11e-01(1.33e-03)+</b>	7.93e-01(4.06e-03)-	2.69e-01(1.30e-01)-	6.68e-01(2.99e-02)-
	8	9.32e-01(1.01e-03)	<b>9.35e-01(1.40e-03)+</b>	9.29e-01(4.41e-03)-	2.01e-01(6.84e-02)-	3.59e-01(1.30e-01)-
	10	<b>9.67e-01(1.32e-03)</b>	9.64e-01(1.61e-03)-	9.65e-01(1.90e-03)-	2.40e-01(3.18e-02)-	1.60e-01(1.00e-01)-
MaF6	5	1.30e-01(6.22e-05)	9.37e-02(4.55e-03)-	1.10e-01(2.78e-03)-	1.27e-01(3.33e-05)-	<b>1.30e-01(1.40e-05)+</b>
	8	<b>1.06e-01(1.66e-05)</b>	9.83e-02(1.15e-02)-	9.50e-02(2.86e-03)-	1.04e-01(3.24e-05)-	9.83e-02(1.26e-03)-
	10	9.22e-02(1.18e-02)	9.12e-02(0.00e+00)~	<b>9.24e-02(1.53e-03)~</b>	1.00e-01(5.94e-06)~	8.06e-02(1.42e-02)-
MaF7	5	2.53e-01(3.05e-02)	<b>3.29e-01(1.70e-03)+</b>	2.95e-01(6.58e-03)+	2.26e-01(2.42e-03)-	2.37e-01(6.73e-03)-
	8	1.94e-01(2.59e-02)	<b>2.89e-01(2.36e-03)+</b>	2.26e-01(6.39e-03)+	2.88e-02(9.26e-02)-	6.78e-02(9.38e-02)-
	10	1.83e-01(3.18e-02)	<b>2.64e-01(1.78e-03)+</b>	2.05e-01(5.98e-03)+	6.37e-03(5.21e-04)-	5.57e-02(1.25e-01)-
Best/all		12/21	7/21	1/21	0/21	1/21
Worse/ similar/better			14-/0~/7+	16-/1~/4+	20-/0~/1+	20-/0~/1+

simultaneously to reach the true PF. The performance of AGPSO is the best on MaF2 with all objectives. As for MaF3, with a large number of local fronts and convex PF, AGPSO and NMPSO are the best in the PSOs described above and AGPSO performs better than NMPSO with 5 objective and worse with 8 and 10 objectives. MaF4, containing a number of local Pareto-optimal fronts, is modified from DTLZ3 by inverting PF shape and AGPSO is better than any PSOs mentioned above in Table 3. Regarding MaF5, modified from DTLZ4, with badly scaled convex PF, AGPSO, NMPSO, and MaPSO solved it all well and the performance of NMPSO is slightly better than AGPSO and MaPSO with 5 and 8 objectives and worse than AGPSO with 10 objectives on MaF5. On MaF6 with a degenerate PF, AGPSO performs best with 8 objectives, while SMPSO performs best with 5 objectives and MaPSO performs best with 10 objectives. Finally, AGPSO does not solve MAF7 with a disconnected PF very well and NMPSO performs best with all objectives.

In the last row of Table 3, SMPSO and dMOPSO perform poorly in solving MaF problems with 5 objectives, especially in the high-dimensional objectives such as 8 and 10 objectives. The main reason is that Pareto dominance used by SMPSO failed in high-dimensional spaces. The pure decomposition-based dMOPSO cannot solve MaOPs very well because a finite fixed reference point does not provide enough search for high-dimensional spaces. AGPSO is better than dMOPSO and SMPSO in 20 and 21 out of 21 cases, respectively, which show superior performance over dMOPSO and SMPSO on MaF problems. Regarding NMPSO, it is competitive with AGPSO, while AGPSO is better than NMPSO in 14 out of 21

cases. Therefore, AGPSO showed the better performance over NMPSO on MaF test problems.

**4.3.2. Comparison Results with Five Current MOEAs.** In the sections mentioned above, it is experimentally proved that AGPSO has better performance compared to existing four MOPSOs on most test problems (MaF and WFG). However, there are not many existing PSO algorithms for dealing with MaOPs. The comparison of PSO algorithms alone does not reflect the superiority of AGPSO; thus, we further compared AGPSO with four current MOEAs (VaEA,  $\theta$ -DEA, MOEA/DD, and SPEA2+SDE).

**(1) Comparison Results on WFG1-WFG9.** The experimental data are listed in Table 4, which shows the HV metric values and the comparison of AGPSO with four competitive MOEAs (VaEA,  $\theta$ -DEA, MOEA/DD, and SPEA2+SDE) on WFG1-WFG9 with 5, 8, and 10 objectives. These four competitive MOEAs are specifically designed for MaOPs, which are better than most of MOPSOs to solve MaOPs; however, experimental results show that they are still worse than AGPSO. As observed from the second last row in Table 4, there are 22 best results in 27 test problems obtained by AGPSO, while MOEA/DD and SPEA2+SDE perform best in 2 and 3 in 27 cases and VaEA and  $\theta$ -DEA are not best on any WFG test problem.

Regarding WFG4, AGPSO performs best with 8 and 10 objectives and slightly worse than MOEA/DD with 5 objectives. For WFG8, MOEA/DD has slightly advantages with 5 objectives compared to AGPSO, but AGPSO shows

TABLE 4: Comparison of results of NMPSTO and four current MOEAs on WF1-WFG9 using HV.

Problem	Obj	AGPSO	VaEA	$\theta$ - DEA	MOEA/DD	SPEA2 + SDE
WFG1	5	<b>7.87e-01(4.27e-02)</b>	3.64e-01(4.02e-02)-	5.73e-01(4.76e-02)-	4.94e-01(2.22e-02)-	7.26e-01(2.27e-02)-
	8	<b>8.75e-01(3.77e-02)</b>	5.78e-01(3.32e-02)-	7.75e-01(3.51e-02)-	3.48e-01(9.75e-02)-	8.36e-01(1.00e-02)-
	10	<b>9.45e-01(3.19e-03)</b>	8.10e-01(3.59e-02)-	8.73e-01(1.59e-02)-	3.40e-01(6.56e-02)-	8.55e-01(8.50e-03)-
WFG2	5	<b>9.76e-01(5.76e-03)</b>	9.62e-01(7.36e-03)-	9.63e-01(6.36e-03)-	9.70e-01(1.14e-03)-	9.60e-01(3.83e-03)~
	8	<b>9.82e-01(4.12e-03)</b>	9.75e-01(6.91e-03)-	9.01e-01(1.65e-01)-	9.42e-01(2.37e-02)-	9.76e-01(3.56e-03)-
	10	<b>9.93e-01(1.99e-03)</b>	9.86e-01(4.59e-03)-	9.34e-01(3.53e-02)-	9.76e-01(1.82e-02)-	9.80e-01(3.09e-03)-
WFG3	5	<b>6.59e-01(5.16e-03)</b>	5.71e-01(1.73e-02)-	6.34e-01(6.13e-03)-	5.69e-01(1.07e-02)-	5.77e-01(2.54e-02)-
	8	<b>6.78e-01(6.20e-03)</b>	5.77e-01(2.83e-02)-	5.69e-01(6.73e-02)-	5.05e-01(1.23e-02)-	5.55e-01(4.06e-02)-
	10	<b>6.91e-01(7.88e-03)</b>	5.63e-01(3.05e-02)-	6.20e-01(1.76e-02)-	4.98e-01(1.55e-02)-	5.42e-01(3.89e-02)-
WFG4	5	7.74e-01(5.34e-03)	7.13e-01(1.06e-02)-	7.61e-01(4.28e-03)-	<b>7.83e-01(4.35e-03)+</b>	7.39e-01(5.10e-03)-
	8	<b>9.00e-01(8.05e-03)</b>	8.05e-01(9.20e-03)-	7.89e-01(2.07e-02)-	6.81e-01(2.59e-02)-	7.85e-01(1.07e-02)-
	10	<b>9.37e-01(6.64e-03)</b>	8.43e-01(9.97e-03)-	8.97e-01(1.14e-02)-	8.53e-01(2.69e-02)-	8.20e-01(8.19e-03)-
WFG5	5	<b>7.41e-01(6.12e-03)</b>	6.97e-01(7.19e-03)-	7.34e-01(3.91e-03)-	7.41e-01(3.39e-03)~	7.08e-01(6.05e-03)-
	8	<b>8.61e-01(4.35e-03)</b>	7.91e-01(1.00e-02)-	7.93e-01(8.22e-03)-	6.66e-01(2.51e-02)-	7.55e-01(1.42e-02)-
	10	<b>8.88e-01(8.29e-03)</b>	8.25e-01(4.03e-03)-	8.69e-01(3.20e-03)-	8.03e-01(1.80e-02)-	7.98e-01(9.63e-03)-
WFG6	5	<b>7.50e-01(8.96e-03)</b>	7.00e-01(1.76e-02)-	7.42e-01(1.06e-02)-	7.40e-01(1.03e-02)-	7.19e-01(1.01e-02)-
	8	<b>8.72e-01(1.26e-02)</b>	8.18e-01(9.27e-03)-	8.16e-01(1.16e-02)-	7.23e-01(2.91e-02)-	7.79e-01(1.13e-02)-
	10	<b>9.09e-01(1.36e-02)</b>	8.49e-01(1.10e-02)-	8.89e-01(1.08e-02)-	8.80e-01(1.29e-02)-	8.17e-01(1.80e-02)-
WFG7	5	<b>7.90e-01(2.12e-03)</b>	7.44e-01(9.10e-03)-	7.90e-01(2.04e-03)~	7.85e-01(3.80e-03)-	7.62e-01(4.15e-03)-
	8	<b>9.20e-01(3.39e-03)</b>	8.71e-01(5.41e-03)-	8.52e-01(1.16e-02)-	7.44e-01(3.15e-02)-	8.22e-01(1.28e-02)-
	10	<b>9.55e-01(9.11e-03)</b>	9.08e-01(5.61e-03)-	9.39e-01(2.22e-03)-	9.22e-01(1.50e-02)-	8.66e-01(6.63e-03)-
WFG8	5	6.63e-01(3.75e-03)	5.98e-01(1.27e-02)-	6.66e-01(5.58e-03)+	<b>6.81e-01(3.20e-03)+</b>	6.60e-01(5.95e-03)-
	8	<b>7.85e-01(7.46e-03)</b>	6.25e-01(2.83e-02)-	6.61e-01(1.85e-02)-	6.72e-01(2.19e-02)-	7.48e-01(1.19e-02)-
	10	<b>8.36e-01(1.30e-02)</b>	6.73e-01(3.66e-02)-	8.02e-01(1.11e-02)-	8.12e-01(7.59e-03)-	7.88e-01(9.94e-03)-
WFG9	5	6.57e-01(3.82e-03)	6.43e-01(1.42e-02)-	6.71e-01(4.34e-02)+	6.55e-01(4.42e-03)-	<b>6.96e-01(8.52e-03)+</b>
	8	7.30e-01(9.96e-03)	7.00e-01(1.12e-02)-	7.12e-01(3.63e-02)-	6.30e-01(3.35e-02)-	<b>7.38e-01(5.00e-02)+</b>
	10	7.49e-01(9.03e-03)	7.20e-01(2.35e-02)-	7.69e-01(5.70e-02)+	6.74e-01(2.56e-02)-	<b>7.75e-01(6.87e-03)+</b>
Best/all		22/27	0/27	0/27	2/27	3/27
Worse/ similar/better			27-0~/0+	23-1~/3+	24-1~/2+	23-1~/3+

excellence in 8 and 10 objectives. As for WFG9,  $\theta$  - DEA and SPEA2 + SDE have more advantages, and AGPSO is worse than  $\theta$  - DEA and SPEA2 + SDE slightly on this test problem, but still better than VaEA and MOEA/DD. For the rest of comparisons on WFG1-WFG3 and WFG5-WFG7, AGPSO has the best performance on most of test problems, which proves the superiority of the proposed algorithm.

In the last row of Table 4, AGPSO performs better than VaEA,  $\theta$  - DEA, MOEA/DD, and SPEA2 + SDE in 27, 23, 24, and 23 out of 27 cases, respectively, while  $\theta$  - DEA, MOEA/DD, and SPEA2 + SDE only have better performance than AGPSO in 3, 2, and 3 out of 27 cases, respectively. In conclusion, AGPSO is found to present a superior performance over four competitive MOEAs in most cases for WFG1-WFG9.

(2) *Comparison Results on MaF1-MaF7.* Table 5 lists comparative results between AGPSO and four competitive MOEAs using HV on MaF1-MaF7 with 5, 8, and 10 objectives. The second last row of Table 5 shows that AGPSO performs best on 11 out of 21 cases, while VaEA,  $\theta$  - DEA, MOEA/DD, and SPEA2 + SDE perform best on 2, 3, 3, 2 cases out of 21 cases, respectively. As a result, AGPSO has a clear advantage over these MOEAs.

Regarding MaF1-MaF2, AGPSO is the best to tackle these test problems in MOEAs mentioned above. For MaF3,

MOEA/DD is the best one and AGPSO has a median performance among the compared MOEAs. Concerning MaF4 with concave PF, AGPSO shows the best performance with all objectives and MOEA/DD performs poorly in the MaF4 test problem. For MaF5, AGPSO only obtained the 3rd rank as it is better than VaEA, MOEA/DD, and SPEA2 + SDE, while outperformed by  $\theta$  - DEA with 5 and 10 objectives and  $\theta$  - DEA with 8 objectives. For MaF6, AGPSO is the best to tackle it with 10 objectives, worse than VaEA with 5 and 8 objectives. As for MaF7, AGPSO performs poorly, while SPEA2 + SDE is the best with 5 and 8 objectives and  $\theta$  - DEA is the best with 10 objectives. As observed from the one-to-one comparisons in the last row of Table 5, AGPSO is better than VaEA,  $\theta$  - DEA, MOEA/DD, and SPEA2 + SDE on 13, 15, 18, and 16 out of 21 cases, respectively, while AGPSO is worse than VaEA,  $\theta$  - DEA, MOEA/DD, and SPEA2 + SDE on 7, 6, 3, and 5 out of 21 cases, respectively.

In conclusion, it is reasonable to conclude that AGPSO shows a better performance over four compared MOEAs in most cases of MaF1-MaF7. This superiority of AGPSO was mainly produced by a novel density-based velocity strategy which enhances the search for areas around each particle and the angle-based external archive update strategy which effectively enhances the performance of AGPSO on MaOPs.

TABLE 5: Comparison of results of AGPSO and four current MOEAs on MaF1-MaF7 using HV.

Problem	Obj	AGPSO	VaEA	$\theta$ - DEA	MOEA/DD	SPEA2 + SDE
MaF1	5	<b>1.31e-02(1.43e-04)</b>	1.11e-02(2.73e-04)-	5.66e-03(2.72e-05)-	2.96e-03(8.34e-05)-	1.29e-02(1.00e-04)-
	8	<b>4.23e-05(1.80e-06)</b>	2.69e-05(2.04e-06)-	2.93e-05(2.03e-06)-	4.38e-06(4.77e-07)-	3.87e-05(9.83e-07)-
	10	<b>6.16e-07(3.49e-08)</b>	3.60e-07(2.21e-08)-	3.39e-07(7.07e-08)-	3.45e-08(2.21e-08)-	5.30e-07(2.16e-08)-
MaF2	5	<b>2.64e-01(1.18e-03)</b>	2.35e-01(4.80e-03)-	2.33e-01(2.82e-03)-	2.34e-01(1.48e-03)-	2.62e-01(1.26e-03)-
	8	<b>2.46e-01(2.64e-03)</b>	2.00e-01(7.59e-03)-	1.71e-01(1.82e-02)-	1.88e-01(1.39e-03)-	2.36e-01(3.06e-03)-
	10	<b>2.45e-01(2.08e-03)</b>	2.01e-01(1.25e-02)-	1.94e-01(7.96e-03)-	1.91e-01(3.21e-03)-	2.30e-01(2.48e-03)-
MaF3	5	9.88e-01(2.87e-03)	9.97e-01(1.13e-03)+	9.93e-01(2.82e-03)+	<b>9.98e-01(9.54e-05)+</b>	9.92e-01(2.12e-03)+
	8	9.95e-01(2.11e-03)	9.98e-01(3.58e-04)+	9.92e-01(3.83e-03)-	<b>1.00e+00(9.42e-07)+</b>	9.97e-01(1.66e-03)+
	10	9.85e-01(2.26e-03)	9.99e-01(9.36e-04)+	9.70e-01(5.68e-03)-	<b>1.00e+00(3.96e-08)+</b>	9.99e-01(4.94e-04)+
MaF4	5	<b>1.29e-01(1.22e-03)</b>	1.17e-01(3.62e-03)-	7.32e-02(8.89e-03)-	0.00e+00(0.00e+00)-	1.07e-01(5.37e-03)-
	8	<b>6.02e-03(9.67e-05)</b>	2.44e-03(3.28e-04)-	1.32e-03(8.49e-04)-	0.00e+00(0.00e+00)-	3.62e-04(7.80e-05)-
	10	<b>5.66e-04(1.89e-05)</b>	1.48e-04(9.76e-06)-	2.29e-04(3.57e-05)-	0.00e+00(0.00e+00)-	4.23e-06(1.65e-06)-
MaF5	5	8.00e-01(2.70e-03)	7.92e-01(1.85e-03)-	<b>8.13e-01(4.51e-05)+</b>	7.73e-01(3.13e-03)-	7.84e-01(4.38e-03)-
	8	<b>9.32e-01(1.01e-03)</b>	9.08e-01(3.80e-03)-	9.26e-01(9.79e-05)-	8.73e-01(6.71e-03)-	7.36e-01(9.62e-02)-
	10	9.67e-01(1.32e-03)	9.43e-01(6.87e-03)-	<b>9.70e-01(4.61e-05)+</b>	9.28e-01(8.28e-04)-	7.12e-01(1.18e-01)-
MaF6	5	1.30e-01(6.22e-05)	<b>1.30e-01(7.51e-05)+</b>	1.17e-01(3.12e-03)-	8.19e-02(2.96e-02)-	1.29e-01(1.84e-04)-
	8	1.06e-01(1.66e-05)	<b>1.06e-01(2.20e-05)+</b>	9.18e-02(1.21e-03)-	2.48e-02(1.29e-02)-	8.81e-02(2.26e-04)-
	10	<b>9.22e-02(1.18e-02)</b>	7.05e-02(3.95e-02)-	4.57e-02(4.66e-02)-	4.83e-02(5.24e-02)-	2.71e-02(1.00e-01)-
MaF7	5	2.53e-01(3.05e-02)	3.03e-01(3.50e-03)+	2.79e-01(7.42e-03)+	1.19e-01(2.76e-02)-	<b>3.31e-01(1.76e-03)+</b>
	8	1.94e-01(2.59e-02)	2.24e-01(6.17e-03)+	2.20e-01(5.03e-02)+	7.06e-04(7.80e-04)-	<b>2.45e-01(1.73e-02)+</b>
	10	1.83e-01(3.18e-02)	1.91e-01(1.19e-02)~	<b>2.27e-01(2.01e-02)+</b>	9.05e-05(4.81e-05)-	6.56e-02(1.05e-02)-
Best/all		11/21	2/21	3/21	3/21	2/21
Worse/ similar/better			13-/1~/7+	15-/0~/6+	18-/0~/3+	16-/0~/5+

*4.4. Further Discussion and Analysis on AGPSO.* To further justify the advantage of AGPSO, it was particularly compared to a recently proposed angle-based evolutionary algorithm VaEA. Their comparison results with HV on MaF and WFG problems on 5 to 10 objectives are already contained in Tables 3 and 5. According the results in the Tables 3 and 5, we can also conclude that AGPSO showed a superior performance over VaEA in most problems of MaF1-MaF7 and WFG1-WFG9, as AGPSO performs better than VaEA in 40 out of 48 comparisons and is only defeated by VaEA in 7 cases. We compare the performance of our algorithm with VaEA's environment selection on different shapes of PF test problems (MaF1-MaF7). They have the same shape of PF but have some different characteristics such as the difficulty of convergence and the deceptive PF property (WFG1-WFG9). Therefore, the angular-guided-based method embedded into PSOs is effectively improved in tackling MaOPs. The angular-guided-based method is adopted as a distributed indicator to distinguish the similarities of particles, which is transformed by the vector angle. The traditional vector angle performs better with concave but worse with convex or linear PFs based on MOEA/C [52]. Therefore, the angular-guided-based method improves this situation; the angular-guided-based method performs better than the traditional vector angle with convex or linear PFs and it will not behave badly with concave PFs. On the one hand, compared with the angle-based strategy designed in VaEA, AGPSO mainly focuses on the diversity first in the update of its external archive and convergence second. On the other hand, AGPSO designed a novel density-based velocity update strategy to produce new particles, which mainly enhances

the search for areas around each particle and speeds up the convergence speed, while balancing convergence and diversity concurrently. From these analyses, we think that our environment selection is more favorable for some linear PFs and concave-shaped PFs with boundaries that are not difficult to find. In addition, AGPSO also added the evolutionary operator on the external archive to improve the performance. Therefore, our proposed AGPSO is reasonable to be regarded as an effective PSO for solving MaOPs.

To visually show and support the abovementioned discussion results, to better visualize the performance, and to show the solutions distribution in high-dimensional objective spaces, some final solution sets with the median HV values from 30 runs were plotted in Figures 2 and 3, respectively, for MaF1 with an inverted PF and for WFG3 with a linear and degenerate PF. In Figure 2, compared with the graph of VaEA, we can find that our boundary cannot be found completely, but the approximate PF we calculated through our proposed AGPSO PF is more closely attached to the real PF. This also proves the correctness of our analysis on environmental selection, i.e., our environment selection is more favorable for some linear PFs and concave-shaped PFs with boundaries that are not difficult to find. The HV trend chart of WFG3, WFG4, and MaF2 with 10 objectives is shown in Figure 4. As observed from Figure 4, the convergence rate of AGPSO is faster than that of the other four optimizers.

*4.5. Further Discussion and Analysis on Velocity Update Strategy.* The abovementioned comparisons have fully proved the superiority of AGPSO over four MOPSOs and four

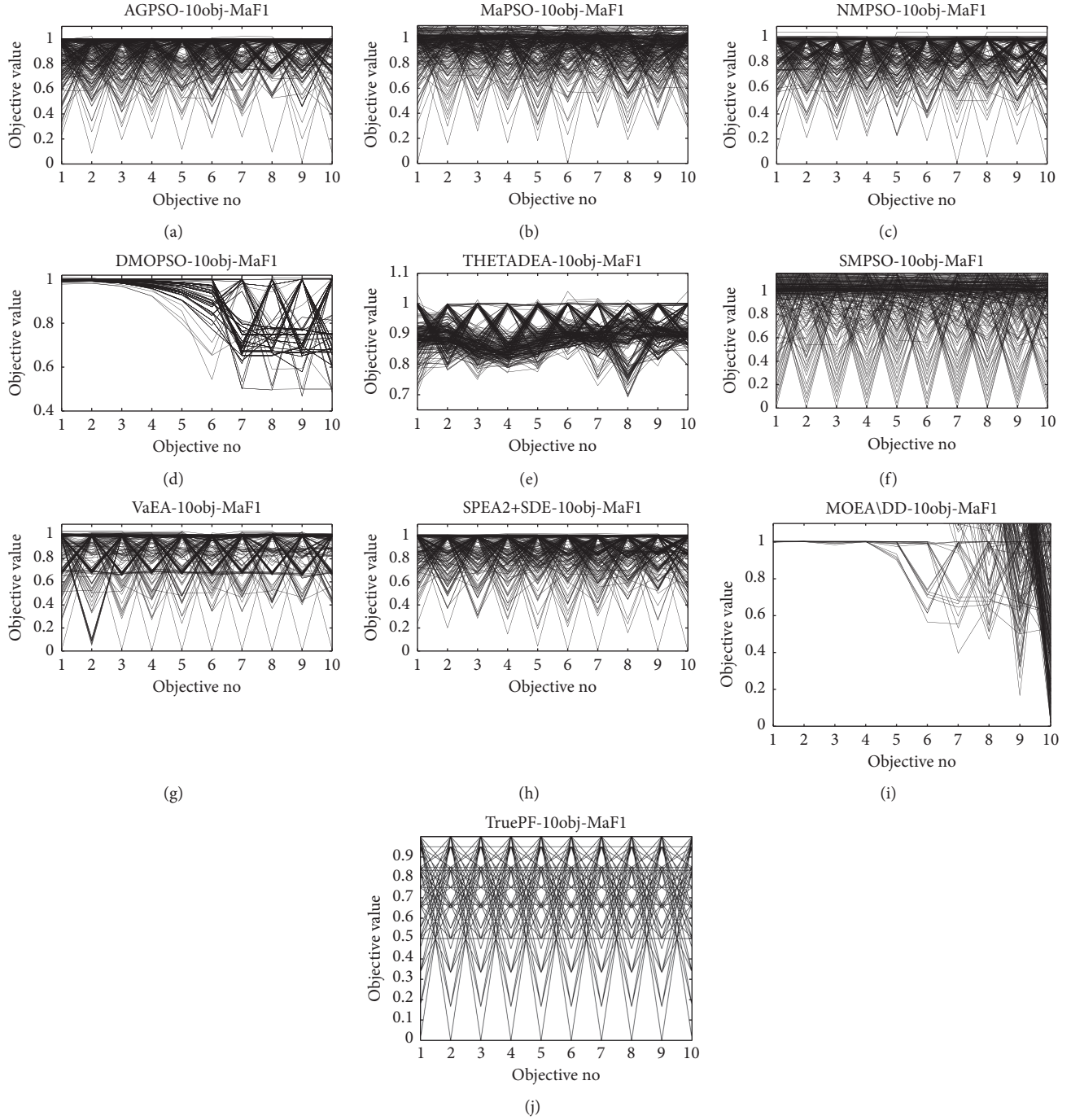


FIGURE 2: The final solution sets achieved by nine MOEAs/MOPSOs and the true PF on MaF1 problems.

MOEAs on the WFG and MaF test problems with 5, 8, and 10 objectives. In this section, we make a deeper comparison of the speed update formula. In order to prove the superiority of the formula based on the density velocity update, the effect is compared with the traditional speed update formula as in [14] (denoted as AGPSO-I). Furthermore, to show that only embedding local best into the traditional velocity update method

$$\begin{aligned}
 v_d(t) = & w_d v_d(t) + c_1 r_{1d} (x_{d,pbest} - x_d(t)) \\
 & + c_2 r_{2d} (x_{d,gbest} - x_d(t)) + c_3 r_{3d} (x_{d,llbest} - x_d(t)),
 \end{aligned} \tag{12}$$

is not sufficient enough, and the effect is also compared with equation (12), which is denoted as AGPSO-II.

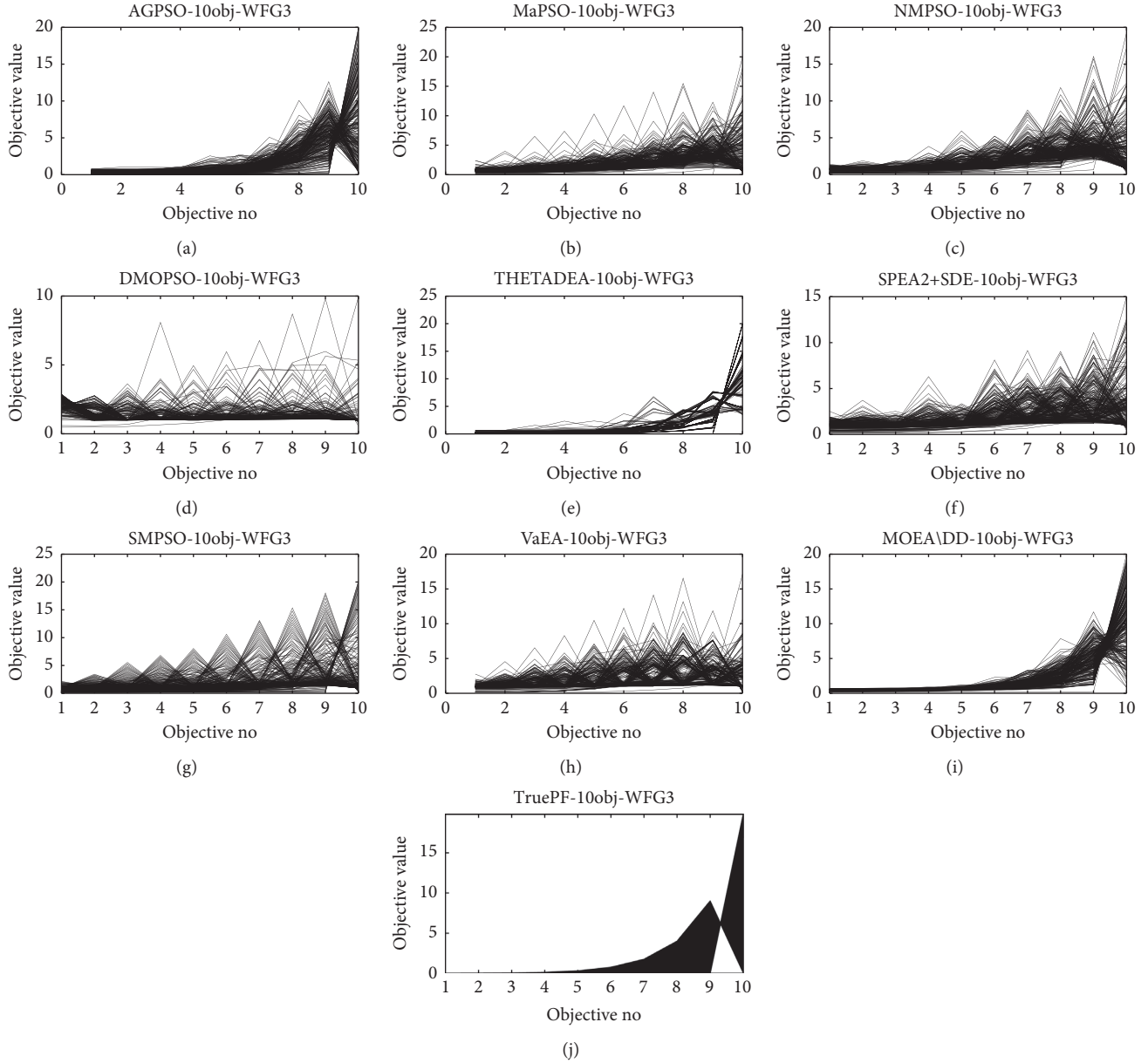


FIGURE 3: The final solution sets achieved by nine MOEAs/MOPSOs and the true PF on WFG3 problems.

Table 6 shows the comparisons of AGPSO and two modified AGPSOs with different speed update formula on WFG1-WFG9 and MaF1-MaF7 using HV. The mean HV values and the standard deviations (included in bracket after the mean HV results) in 30 runs are collected for comparison. As in the second last row of Table 6, there are 34 best results in 48 test problems obtained by AGPSO, while AGPSO-I performs best in 10 cases and AGPSO-II performs best only in 4 case, respectively. Regarding the comparison of AGPSO with AGPSO-I, AGPSO performs better on 33 cases, similarly on 5 cases, and worse on 10 cases. The novel velocity update strategy based on density is effective in improving the performance of AGPSO on WFG4, WFG5, WFG7, WFG9, and MaF2-MaF6. From these comparison data, the novel velocity update strategy can improve the AGPSO and enable particle generation to be more efficient under the coordination of the

proposed angular-guided update strategies. Regarding the comparison AGPSO and AGPSO-II, AGPSO performs better on 31 cases, similarly on 12 cases, and worse on 5 cases. It also verified the superiority of the proposed novel velocity update strategy by comparing the variants of the traditional formula defined by (12). Adding the local-best (*lbest*) positional information of a particle to the traditional velocity formula is not enough to control the search strength in the surrounding region of this particle. It also confirmed that the proposed formula can produce higher quality particles.

**4.6. Computational Complexity Analysis on AGPSO.** The computational complexity of AGPSO in one generation is mainly dominated by the environmental selection that is described in Algorithm 2. Algorithm 2 requires a time complexity

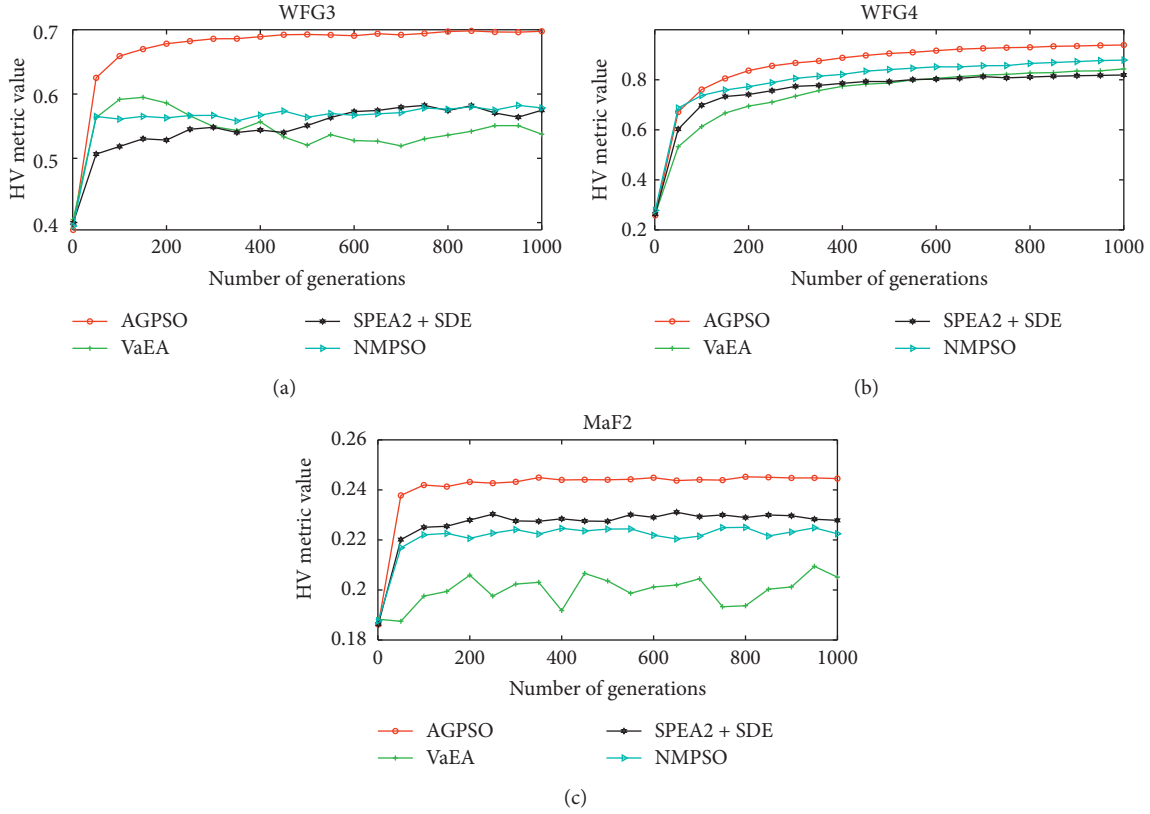


FIGURE 4: The HV trend charts (averaged over 30 runs) of AGPSO, SPEA2 + SDE, and NMPSO on WFG3, WFG4, and MaF2 with 10 objectives.

TABLE 6: Comparison of results of AGPSO with different velocity using HV.

Problem	Obj	AGPSO	AGPSO-I	AGPSO-II
WFG1	5	<b>7.91e-01</b> (3.47e-02)	7.48e-01(3.41e-02)-	7.51e-01(3.88e-02)-
	8	<b>8.85e-01</b> (2.54e-02)	8.14e-01(3.94e-02)-	7.96e-01(3.97e-02)-
	10	9.47e-01(4.77e-03)	<b>9.49e-01</b> (2.59e-03)+	9.45e-01(3.49e-03)~
WFG2	5	9.76e-01(5.13e-03)	9.76e-01(2.81e-03)~	<b>9.78e-01</b> (5.19e-03)+
	8	9.87e-01(2.65e-03)	<b>9.89e-01</b> (2.21e-03)+	9.88e-01(1.84e-03)~
	10	9.92e-01(3.30e-03)	<b>9.93e-01</b> (1.38e-03e-)+	9.92e-01(1.97e-03)~
WFG3	5	6.59e-01(5.02e-03)	<b>6.66e-01</b> (4.17e-03)+	6.57e-01(5.40e-03)~
	8	6.79e-01(5.20e-03)	<b>6.86e-01</b> (3.41e-03)+	6.79e-01(4.02e-03)~
	10	6.92e-01(7.15e-03)	<b>7.02e-01</b> (3.08e-03)+	6.93e-01(4.13e-03)~
WFG4	5	<b>e-7.76e-01</b> (5.69e-03)	7.57e-01(7.87e-03)-	7.68e-01(5.07e-03)-
	8	<b>9.00e-01</b> (1.01e-02)	8.85e-01(8.62e-03)-	8.94e-01(8.43e-03)-
	10	<b>9.39e-01</b> (7.42e-03)	9.31e-01(8.35e-03)-	9.30e-01(8.84e-03)-
WFG5	5	<b>7.42e-01</b> (4.66e-03)	7.34e-01(7.45e-03)-	7.39e-01(7.73e-03)-
	8	<b>8.59e-01</b> (4.74e-03)	8.52e-01(6.05e-03)-	8.58e-01(5.45e-03)~
	10	<b>8.91e-01</b> (5.28e-03)	8.73e-01(1.22e-02)-	8.78e-01(2.21e-02)-
WFG6	5	<b>7.53e-01</b> (7.52e-03)	6.92e-01(2.21e-03)-	7.52e-01(3.85e-03)~
	8	8.69e-01(1.24e-02)	8.10e-01(3.63e-03)-	<b>8.83e-01</b> (5.64e-03)+
	10	9.12e-01(1.22e-02)	8.40e-01(3.57e-03)-	<b>9.23e-01</b> (7.99e-03)+
WFG7	5	<b>7.90e-01</b> (2.13e-03)	7.81e-01(3.32e-03)-	7.85e-01(2.78e-03)-
	8	<b>9.21e-01</b> (2.89e-03)	9.15e-01(3.11e-03)-	9.13e-01(3.18e-03)-
	10	<b>9.55e-01</b> (9.72e-03)	9.55e-01(2.26e-03)~	9.44e-01(1.59e-02)-
WFG8	5	<b>6.61e-01</b> (5.62e-03)	6.44e-01(3.76e-03)-	6.54e-01(5.95e-03)-
	8	<b>7.87e-01</b> (7.11e-03)	7.80e-01(4.75e-03)-	7.81e-01(5.89e-03)-
	10	8.41e-01(5.48e-03)	<b>8.45e-01</b> (3.13e-03)+	8.31e-01(1.76e-02)-

TABLE 6: Continued.

Problem	Obj	AGPSO	AGPSO-I	AGPSO-II
WFG9	5	<b>6.58e-01(4.54e-03)</b>	6.53e-01(3.73e-03)-	6.53e-01(2.99e-03)-
	8	<b>7.29e-01(9.55e-03)</b>	7.20e-01(1.21e-02)-	7.23e-01(8.94e-03)-
	10	<b>7.51e-01(1.21e-02)</b>	7.35e-01(1.18e-02)-	7.36e-01(1.04e-02)-
MaF1	5	1.30e-02(1.29e-04)	1.30e-02(1.10e-04)~	<b>1.30e-02(1.04e-04)~</b>
	8	<b>4.23e-05(1.45e-06)</b>	4.12e-05(1.35e-06)-	4.14e-05(1.78e-06)-
	10	<b>6.12e-07(4.40e-08)</b>	6.07e-07(2.30e-08)~	6.10e-07(2.83e-08)~
MaF2	5	<b>2.64e-01(9.97e-04)</b>	2.62e-01(1.08e-03)-	2.61e-01(1.10e-03)-
	8	<b>2.46e-01(1.64e-03)</b>	2.40e-01(1.89e-03)-	2.41e-01(1.66e-03)-
	10	<b>2.45e-01(1.84e-03)</b>	2.39e-01(2.75e-03)-	2.41e-01(2.19e-03)-
MaF3	5	<b>9.90e-01(4.96e-03)</b>	9.76e-01(2.92e-02)-	4.96e-01(3.82e-01)-
	8	<b>9.68e-01(1.95e-03)</b>	9.36e-01(9.22e-02)-	3.73e-01(3.08e-01)-
	10	<b>9.92e-01(2.55e-03)</b>	9.72e-01(1.86e-02)-	4.42e-01(3.77e-01)-
MaF4	5	<b>1.29e-01(1.33e-03)</b>	1.08e-01(1.44e-02)-	1.18e-01(2.08e-03)-
	8	<b>6.03e-03(1.88e-04)</b>	3.33e-03(1.02e-03)-	4.74e-03(2.12e-04)-
	10	<b>5.62e-04(1.47e-05)</b>	2.58e-04(9.13e-05)-	3.65e-04(2.55e-05)-
MaF5	5	<b>8.01e-01(1.81e-03)</b>	8.00e-01(3.40e-03)~	8.00e-01(2.06e-03)~
	8	<b>9.32e-01(1.97e-03)</b>	9.31e-01(1.39e-03)-	9.31e-01(1.58e-03)-
	10	<b>9.67e-01(1.09e-03)</b>	9.66e-01(7.51e-04)-	9.66e-01(1.18e-03)-
MaF6	5	<b>1.30e-01(5.82e-05)</b>	1.30e-01(3.35e-05)-	1.30e-01(5.60e-05)-
	8	<b>1.03e-01(2.78e-05)</b>	4.51e-02(4.39e-02)-	6.64e-02(5.81e-02)-
	10	<b>9.49e-02(1.27e-02)</b>	4.55e-02(7.07e-02)-	4.40e-02(4.92e-02)-
MaF7	5	2.51e-01(4.40e-02)	<b>3.09e-01(2.58e-03)+</b>	2.49e-01(3.73e-02)~
	8	1.91e-01(4.11e-02)	<b>2.54e-01(4.64e-03)+</b>	2.22e-01(1.68e-02)+
	10	1.78e-01(3.53e-02)	<b>2.32e-01(4.46e-03)+</b>	2.01e-01(9.95e-03)+
Best/all	34/48	10/48	4/48	
Better/similar/worse		33-/5~/10+	31-/12~/5+	

of  $O(mN)$  to obtain the union population  $U$  in line 1 and normalize each particle in  $U$  in line 2, where  $m$  is the number of objectives and  $N$  is the swarm size. In lines 3–7, it requires a time complexity of  $O(m^2N^2)$  to classify the population. In lines 8–11, it needs time complexity of  $O(N)$ . In the loop for association, it also requires a time complexity of  $O(m^2N^2)$  as in lines 12–15. In lines 17–20, it needs a time complexity of  $O(N^2)$ . In conclusion, the overall worst time complexity of one generation in AGPSO is  $\max\{O(mN), O(m^2N^2), O(N), O(N^2)\}$ , which performs competitive time complexities with most optimizers, e.g., [48].

## 5. Conclusions and Future Work

This paper proposes AGPSO, which is a novel angular-guided MOPSO with efficient density-based velocity update strategy and excellent angular-guided-based archive update strategy. The novel density-based velocity update strategy uses adjacent information (local-best) to explore information around particles and uses globally optimal information (global-best) to search for better performing locations globally. This strategy improves the quality of the particles produced by searching for the surrounding space of sparse particles. Furthermore, the angular-guided archive update strategy provides efficient convergence while maintaining good population distribution. This combination of proposed novel velocity update strategy and excellent archive update strategy enables the proposed AGPSO to overcome the problems encountered in the existing state-of-the-art algorithms for solving MaOPs. The performance of

AGPSO is assessed by using WFG and MaF test suites with 5 to 10 objectives. Our experiments indicate that AGPSO has superior performance over four current PSOs (SMPSO, dMOPSO, NMP SO, and MaPSO) and four evolutionary algorithms (VaEA,  $\theta$ -DEA, MOEA/D-DD, and SPEA2 + SDE).

This density-based velocity update strategy and angular-guided archive update strategy will be further studied in our future work. The performance of density velocity update strategy is not good enough on some partial problems, which will be further studied. Regarding angular-guided archive update strategy, while reducing the amount of computation and performing better than the tradition angle selection on concave and linear PFs, it also sacrifices a part of the performance against many objective optimization problems with convex PFs, the improvement of which will be further studied in future.

## Data Availability

Our source code could be provided by contacting the corresponding author. The source codes of the compared state-of-the-art algorithms can be downloaded from <http://jmetal.sourceforge.net/index.html> or provided by the original author, respectively. Also, most codes of the compared algorithm can be found in our source code, such as VaEA [27],  $\theta$ -DEA [48], and MOEA/DD [50]. Test problems are the WFG [23] and MaF [24]. WFG [23] can be found at <http://jmetal.sourceforge.net/problems.html> and MaF [24] can be found at <https://github.com/BIMK/PlatEMO>.



## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61876110, 61872243, and 61702342, Natural Science Foundation of Guangdong Province under Grant 2017A030313338, Guangdong Basic and Applied Basic Research Foundation under Grant 2020A151501489, and Shenzhen Technology Plan under Grants JCYJ20190808164211203 and JCYJ20180305124126741.

## References

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [3] Q. F. Zhang and H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [4] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [5] W. Hu and G. G. Yen, "Adaptive multi-objective particle swarm optimization based on parallel cell coordinate system," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 1–18, 2015.
- [6] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Washington, DC, USA, June 1995.
- [7] M. Gong, L. Jiao, H. Du, and L. Bo, "Multi-objective immune algorithm with nondominated neighbor-based selection," *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.
- [8] R. C. Eberhart and Y. H. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 81–86, Seoul, Republic of Korea, May 2001.
- [9] W. Zhang, G. Li, W. Zhang, J. Liang, and G. G. Yen, "A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 50, p. 100569. In press, 2019.
- [10] D. Tian and Z. Shi, "MPSO: modified particle swarm optimization and its applications," *Swarm and Evolutionary Computation*, vol. 41, pp. 49–68, 2018.
- [11] S. Rahimi, A. Abdollahpouri, and P. Moradi, "A multi-objective particle swarm optimization algorithm for community detection in complex networks," *Swarm and Evolutionary Computation*, vol. 39, pp. 297–309, 2018.
- [12] J. García-Nieto, E. López-Camacho, García-Godoy, M. J. Nebro, A. J. Nebro, and J. F. Aldana-Montes, "Multi-objective ligand-protein docking with particle swarm optimizers," *Swarm and Evolutionary Computation*, vol. 44, pp. 439–452, 2019.
- [13] M. R. Sierra and C. A. Coello Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance," in *Evolutionary Multi-Criterion Optimization*, vol. 3410 of Lecture Notes in Computer Science, pp. 505–519, Springer, Berlin, Germany, 2005.
- [14] A. J. Nebro, J. J. Durillo, J. García-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "SMPSO: a new PSO-based meta-heuristic for multi-objective optimization," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 66–73, Nashville, TN, USA, March 2009.
- [15] Z. Zhan, J. Li, J. Cao, J. Zhang, H. Chuang, and Y. Shi, "Multiple populations for multiple objectives: a coevolutionary technique for solving multi-objective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.
- [16] H. Han, W. Lu, L. Zhang, and J. Qiao, "Adaptive gradient multiobjective particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3067–3079, 2018.
- [17] S. Zapotecas Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 69–76, Dublin, Ireland, July 2011.
- [18] N. Al Moubayed, A. Pertovski, and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evolutionary Computation*, vol. 22, no. 1, pp. 47–77, 2014.
- [19] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming, "A novel multi-objective particle swarm optimization with multiple search strategies," *European Journal of Operational Research*, vol. 247, no. 3, pp. 732–744, 2015.
- [20] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [21] D. H. Phan and J. Suzuki, "R2-ibea: R2 indicator based evolutionary algorithm for multi-objective optimization," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 1836–1845, Cancun, Mexico, July 2013.
- [22] S. Jia, J. Zhu, B. Du, and H. Yue, "Indicator-based particle swarm optimization with local search," in *Proceedings of the 2011 Seventh International Conference on Natural Computation*, vol. 2, pp. 1180–1184, Shanghai, China, July 2011.
- [23] X. Sun, Y. Chen, Y. Liu, and D. Gong, "Indicator-based set evolution particle swarm optimization for many-objective problems," *Soft Computing*, vol. 20, no. 6, pp. 2219–2232, 2015.
- [24] Q. Lin, S. Liu, Q. Zhu et al., "Particle swarm optimization with A balanceable fitness estimation for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32–46, 2018.
- [25] 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.
- [26] F. Li, J. C. Liu, S. B. Tan, and X. Yu, "R2-MOPSO: a multi-objective particle swarm optimizer based on R2-indicator and decomposition," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3148–3155, Sendai, Japan, May 2015.
- [27] K. Narukawa and T. Rodemann, "Examining the performance of evolutionary many-objective optimization algorithms on a real-world application," in *Proceedings of the Sixth*

- International Conference on Genetic and Evolutionary Computing*, pp. 316–319, Kitakushu, Japan, August 2012.
- [28] F. López-Pires, “Many-objective resource allocation in cloud computing datacenters,” in *Proceedings of the 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, pp. 213–215, Berlin, Germany, April 2016.
- [29] 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 Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 587–602. In press, 2019.
- [30] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.
- [31] R. Cheng, M. Li, Y. Tian et al., “A benchmark test suite for evolutionary many-objective optimization,” *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.
- [32] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, “Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.
- [33] X. He, Y. Zhou, and Z. Chen, “An evolution path based reproduction operator for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 29–43, 2017.
- [34] N. Lynn, M. Z. Ali, and P. N. Suganthan, “Population topologies for particle swarm optimization and differential evolution,” *Swarm and Evolutionary Computation*, vol. 39, pp. 24–35, 2018.
- [35] Q. Zhu, Q. Lin, W. Chen et al., “An external archive-guided multiobjective particle swarm optimization algorithm,” *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2794–2808, 2017.
- [36] W. Hu, G. G. Yen, and G. Luo, “Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system,” *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1446–1459, 2017.
- [37] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multiobjective optimization,” *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
- [38] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, “A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 331–345, 2019.
- [39] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, “Hyperplane assisted evolutionary algorithm for many-objective optimization problems,” *IEEE Transactions on Cybernetics*, pp. 1–14. In press, 2019.
- [40] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [41] X. He, Y. Zhou, Z. Chen, and Q. Zhang, “Evolutionary many-objective optimization based on dynamical decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 361–375, 2019.
- [42] F. Gu and Y.-M. Cheung, “Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2018.
- [43] X. Cai, H. Sun, and Z. Fan, “A diversity indicator based on reference vectors for many-objective optimization,” *Information Sciences*, vol. 430–431, pp. 467–486, 2018.
- [44] T. Pamulapati, R. Mallipeddi, and P. N. Suganthan, “I<sub>SDE+</sub>—an indicator for multi and many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 346–352, 2019.
- [45] Y. N. Sun, G. G. Yen, and Z. Yi, “IGD indicator-based evolutionary algorithm for many-objective optimization problem,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2019.
- [46] Y. Xiang, Y. Zhou, Z. Chen, and J. Zhang, “A many-objective particle swarm optimizer with leaders selected from historical solutions by using scalar projections,” *IEEE Transactions on Cybernetics*, pp. 1–14. In press, 2019.
- [47] M. Li, S. Yang, and X. Liu, “Shift-based density estimation for Pareto-based algorithms in many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.
- [48] Y. Xiang, Y. R. Zhou, and M. Q. Li, “A vector angle based evolutionary algorithm for unconstrained many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.
- [49] K. Li, K. Deb, Q. F. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [50] S. Huband, L. Barone, R. while, and P. Hingston, “A scalable multi-objective test problem toolkit,” in *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization*, pp. 280–295, vol. 3410 of Lecture Notes in Computer Science, Guanajuato, Mexico, March 2005.
- [51] Q. Lin, S. Liu, K. Wong et al., “A clustering-based evolutionary algorithm for many-objective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405. In press, 2018.
- [52] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” in *Evolutionary Multi-Objective Optimization, Ser. Advanced Information and Knowledge Processing*, A. Abraham, L. Jain, and R. Goldberg, Eds., pp. 105–145, Springer London, London, UK, 2005.