Multi-Exemplar Particle Swarm Optimization

WEI SONG\(^1,2\), (Member, IEEE), AND ZIYU HUA\(^1\)

\(^1\)School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
\(^2\)Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China

Corresponding author: Wei Song (songwei@jiangnan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673193 and Grant 62076110, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20181341, and in part by the China Postdoctoral Science Foundation under Grant 2017M621625.

ABSTRACT PSO and its variants have proven to be useful algorithms for tackling a wide range of optimization problems in recent decades. However, PSO and most of its variants only consider the influences caused by global best position and personal historical best position. Such a single way of influencing often leads to an issue of insufficient diversity of population, and further makes the algorithms prone to falling into local optimum. In this paper, we propose a multi-exemplar particle swarm optimization (MEPSO) to deal with this issue. Specifically, each particle will choose the global best particle and the best companion particle as its exemplars, which brings more useful knowledge for particle update. To further describe the influences with respect to different exemplars, we define two influence coefficients inspired by mechanics. Such influence coefficients ensure that the best current experience is shared while enrich the diversity of population. Moreover, in the light of the distance between each particle and its best companion particle on each dimension, a variable-scale search is given in this paper to enhance the overall convergence ability. To verify the effectiveness of our algorithm, we conduct abundant experiments on all functions of CEC2013 test suite. The experimental results show that MEPSO performs better than 14 competitors in terms of comprehensive performance and achieves state-of-the-art results.

INDEX TERMS Particle swarm optimization, multi-exemplar, influence coefficient, variable-scale search.

I. INTRODUCTION

Inspired by swarm behaviors of animals in nature, Particle Swarm Optimization (PSO), as one of representative swarm intelligence algorithms, was proposed by Kennedy and Eberhart in 1995 [1]. Due to its strong universality, PSO has attracted extensive attention in recent decades. As PSO is easy to put into practice, it has been successfully applied to various fields such as machine design [2], [3], neural networks [4], [5], path planning [6], [7] and etc.

In original PSO, two acceleration coefficients with respect to the influences of global best position and personal historical best position play a crucial role to determine the search trajectory of particle and control the search ability. To further balance the global and local search, Shi and Eberhart proposed a canonical PSO with a new parameter named inertia weight [8]. Then they conducted an empirical study on the linearly decreasing inertia weight from 0.9 to 0.4, considering the extent and accuracy of search in some extent [9]. Currently, the common methods for updating the inertia weight still follow the linearly decreasing rule, aiming to meet search requirements in different evolutionary stages [10], [11]. Benefited from the idea of time-varying inertia weight, a series of PSO variants with time-varying acceleration coefficients were set forth [12], [13]. In addition, the topological adjustments based on iteration number enhance the performance of PSO in some extent [14], [15]. However, these improvements just considering iteration number to adjust PSO may result in the lack of wisdom for particles, especially in the face of complex tasks. Moreover, at the end of search, the global search ability will be lost. Differing from the improvements just by using iteration number, some PSO variants employ fitness to adjust parameters [16], [17] and learning models [18], [19]. These fitness-based improvements assign effective controls to each particle, forcing the particles in swarm to play different roles in dealing with complex tasks.

In this paper, we propose a novel multi-exemplar particle swarm optimization to expand the diversity of the population and enhance the convergence ability. Specifically, for each particle we first select the global best particle and the best companion particle as its exemplars, which brings...
more useful knowledge for particle update. To describe the influences with respect to different exemplars, we define two influence coefficients inspired by mechanics. Such influence coefficients ensure that the best current experience is shared while enrich the diversity of population. Moreover, in terms of the distance between each particle and its best companion particle on each dimension, the variable-scale search is given in this paper to enhance the overall convergence ability.

The rest of this paper is organized as follows. Section II introduces the canonical PSO and its variants. The details of MEPSO are described in Section III. In Section IV, we conduct abundant experiments by comparing MEPSO with the canonical PSO, 10 state-of-art PSO variants and 3 representative non-PSO optimization algorithms. The experimental results show that MEPSO achieves better comprehensive performance. Finally, we conclude the study in Section V.

II. RELATED WORK
A. CANONICAL PSO
In the canonical PSO, the position of each particle represents the motive of movement. In a D dimensional space, suppose the velocity vector and position vector of the ith particle are represented by \( V_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \) and \( X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \) respectively, they will be updated in each iteration according to Eqs. (1) and (2).

\[
\begin{align*}
\dot{v}_{id}(t + 1) &= w \times v_{id}(t) + c_1 \times r_1 \times (p_{bestd}(t) - x_{id}(t)) + c_2 \times r_2 \times (g_{bestd}(t) - x_{id}(t)) \\
\dot{x}_{id}(t + 1) &= x_{id}(t) + v_{id}(t)
\end{align*}
\]

where \( t \) and \( t + 1 \) represent the \( t \)th and the \( (t + 1) \)th generations respectively, \( d \) refers to the \( d \)th dimension of each particle. \( r_1 \) and \( r_2 \) are two independent random variables following uniform distribution in range \([0, 1]\). \( c_1 \) and \( c_2 \) are two acceleration coefficients, and \( w \) is an inertia weight. \( p_{best} = (p_{best1}, p_{best2}, \ldots, p_{bestD}) \) represents the personal best position achieved so far for the ith particle, and \( g_{best} = (g_{best1}, g_{best2}, \ldots, g_{bestD}) \) is the global best position found in the whole swarm.

In canonical PSO, particles are first initialized with a population of random positions in the solution space. Then, each particle updates its velocity and position based on Eqs. (1) and (2) in the way of iteration. Note that, during the search process, the personal best fitness and personal best position of each particle are maintained. Once a particle finds a better solution, the corresponding fitness and position are utilized to update the personal best fitness and the personal best position respectively. Similarly, the swarm will update global best position and global best fitness when a better global best solution is found.

B. PSO VARIANTS BASED ON VARIOUS LEARNING MODELS
To enhance the learning ability of swarm, a series of PSO variants were developed based on different topology spaces and learning approaches.

It has been verified by Kennedy and Mendes [20] that a smaller neighborhood is proper for handling multimodal problems while a larger neighborhood can perform much better on unimodal problems. Thus, to overcome the shortcoming of fixed neighborhoods, some dynamic neighborhood update strategies were studied for finding the suitable topology of neighborhood. For instance, Liang et al. [21] proposed a Comprehensive Learning Particles Swarm Optimization (CLPSO) to find high quality solutions for multimodal functions. In CLPSO, each particle is allowed to learn from the personal best position of a randomly selected particle. In comparison with the canonical PSO, the variants based on the given topology spaces maintain the swarm diversity and avoid premature convergence. However, the individuals in the same neighborhood usually have similar information, which narrows the information diffusion within the population. In addition, frequent topology adjustment will inevitably increase extra computational cost.

On the other hand, Xu et al. [22] try to overcome the phenomena of “two steps forward, one step back”. They focused on the dimensional information of global best particle and proposed a dimensional learning approach. In such an approach, particles learn effective information from each dimension of the global best particle, aiming to improve the convergence ability. However, all particles in swarm overly depend on the experience of the global best particle, causing an issue of insufficient diversity in population. To deal with this issue, Xia et al. proposed a XPSO with a forgetting ability [23]. It reduces the amount of information obtained from the global best particle, aiming to enrich the population diversity. However, the convergence ability of XPSO will be reduced just by enricbing the diversity of population.

In contrast, MEPSO is proposed in this paper to expand the diversity of the population and enhance the convergence ability. Specifically, the multi-exemplar learning based on influence coefficient is first given mainly for enriching the diversity of population. Moreover, a variable-scale search is proposed based on the distance between each particle and its best companion particle on each dimension to improve the overall convergence ability of MEPSO.

III. MULTI-EXEMPLAR PARTICLE SWARM OPTIMIZATION (MEPSO)
A. MULTI-EXEMPLAR MODEL
In the canonical PSO, the global best particle is regarded as the unique elite of “social learning” part. However, just learning from the global best particle will force the whole population to approach that particle, which may reduce the diversity of the swarm. In this study, we propose a multi-exemplar model, in which the global best particle is considered as one of the exemplars for each particle. In fact, the particle swarm can be seen as the human society, in which learning from multiple exemplars is more effective, and such a learning mode has become a common phenomenon in human society. It has been verified by many studies that individual with multi-exemplars can obtain a better learning
capability [24], [25]. Whereas, learning from too many exemplars will bring learning into chaos, influencing the efficiency of learning. Thus, besides the global best particle (gbest), best companion particle (cbest) is also considered as one of the exemplars in multi-exemplar model. Specifically, the companion particles are randomly selected from the current swarm. Differing from selecting the leaning object in the same neighborhood topology, in this paper each particle selects its companions randomly, which can bring manifold information.

### B. MULTI-EXEMPLAR LEARNING BASED ON INFLUENCE COEFFICIENT

In recent years, researchers have attempted to utilize gravity model to explain various sociology phenomena from the perspective of mechanics [26]. In this paper, to further describe the influences with respect to different exemplars, we define two influence coefficients inspired by mechanics. Firstly, to clearly reflect the impact of gbest and pbest, we remove the independent random variables \( r_1 \) and \( r_2 \), the acceleration coefficients \( c_1 \) and \( c_2 \), and inertia weight \( w \) in Eq. (1). The simplified velocity update formula is given as

\[
v_i(t+1) = v_i(t) + \alpha_i d_i(t) + \beta_i (pbest_i(t) - x_i(t)) + \gamma_i (gbest_i(t) - x_i(t))
\]

(3)

As seen from Eq. (3), it is obvious that the motion of each particle will be affected by gbest and pbest. According to Newton’s first law of motion, force is the unique factor to change the motion of object. Thus, the two vectors, i.e., \( pbest_i(t) - x_i(t) \) and \( gbest_i(t) - x_i(t) \), in Eq. (3) can be seen as two forces. That is because they affect the motion of each particle. Since pbest reflects the learning experience of each individual, and a series of pbests in population can be regarded as the swarm diversity. Differing from pbest, gbest reflects the best learning experience in population, and it affects all individuals as a primary factor. Based on Eq. (3), the farther an individual is from gbest, the greater the force will be, causing a decline of swarm diversity. To address this issue, we define a global influence coefficient \( ggr_i \) and a companion influence coefficient \( cgr_i \) to describe the influences of the gbest and cbest on the \( i \)th particle, respectively, from the perspective of mechanics

\[
ggr_i = \left( \frac{\text{dist}_{g_{\text{max}}}^2}{\text{dist}_{g_{\text{max}}}^2 + \text{dist}_{g_i}^2} \right)
\]

(4)

\[
cgr_i = \left( \frac{\text{dist}_{c_{\text{max},i}}^2}{\text{dist}_{c_{\text{max},i}}^2 + \text{dist}_{c_i}^2} \right)
\]

(5)

where \( \text{dist}_{g_{\text{max}}} \) is the distance between the global best particle and its farthest particle. \( \text{dist}_{c_{\text{max},i}} \) represents the distance between the best companion and its farthest companion. Subsequently, \( \text{dist}_{g_{\text{max}}} \) and \( \text{dist}_{c_{\text{max},i}} \) are respectively calculated by

\[
\text{dist}_{g_{\text{max}}} = \sqrt{\sum_{d=1}^{D} (gbest_d - x_{\text{farthest},d})^2}
\]

(6)

In Eqs. (4) and (5), \( \text{dist}_{g_i} \) and \( \text{dist}_{c_i} \) represent the distances between the \( i \)th particle and its two exemplars. Subsequently, \( \text{dist}_{g_i} \) and \( \text{dist}_{c_i} \) are respectively calculated by

\[
\text{dist}_{g_i} = \sqrt{\sum_{d=1}^{D} (x_i^d - gbest^d)^2}
\]

(8)

\[
\text{dist}_{c_i} = \sqrt{\sum_{d=1}^{D} (x_i^d - cbest^d)^2}
\]

(9)

Because \( \text{dist}_{g_{\text{max}}} \) is the upper limit of \( \text{dist}_{g_i} \), based on Eq. (4), it is obvious that \( ggr_i \) belongs to the range of \([0.5,1]\). In addition, as the \( i \)th particle and its best companion particle choose their companions separately, There is no direct relation between \( \text{dist}_{c_i} \) and \( \text{dist}_{c_{\text{max},i}} \). Thus, based on Eq. (5) \( cgr_i \) is in the interval of \([0,1]\). In MEPSO, the global best particle is used to share the best experience of the swarm to each particle. Thus, we set \( ggr_i \) with a relatively large value \((0.5 \leq ggr_i \leq 1)\) to ensure that the best experience can be learned. Differing from the global best particle, the best companion particle is utilized to expand the diversity of swarm. Thus, we set the range of \( cgr_i \) as \( 0 < cgr_i < 1 \). Based on Eq. (5), the smaller the distance between the \( i \)th particle and its best companion particle, the greater the \( cgr_i \). Moreover, in contrast to the canonical PSO, \( ggr_i \) is utilized to reduce the influence of the global best particle to some extent and makes the best experience of swarm can be shared. More importantly, we use \( cgr_i \) to enrich the diversity of swarm. After defining \( ggr_i \) and \( cgr_i \), the velocity of each particle is updated as

\[
v_i(t+1) = \omega \times v_i(t) + c_1 \times r_1 \times (pbest_i(t) - x_i(t)) + ggr_i \times c_2 \times r_2 \times (gbest_i(t) - x_i(t)) + cgr_i \times c_3 \times r_3 \times (cbest_i(t) - x_i(t))
\]

(10)

where \( cbest_i \) represents the personal best position of the best companion particle. Similar to Eq. (1), \( r_1, r_2 \) and \( r_3 \) are three independent random variables in range \([0,1]\). \( c_1, c_2 \) and \( c_3 \) are three acceleration coefficients.

### C. VARIABLE-SCALE SEARCH BASED ON BEST COMPANION

In this paper, the multi-exemplar learning based on influence coefficient is firstly proposed mainly for expanding the diversity of swarm, which increases the possibility of MEPSO converging to global optimum. To make the algorithm converge to the global optimum, based on the distance between each particle and its best companion particle on each dimension, a variable-scale search is further proposed to improve the overall convergence ability of MEPSO.

In the research field of swarm intelligence, crisscross optimization algorithm (CSO) is regarded as a “catalytic agent” that has the powerful ability to search for desired solution.
with high quality. In CSO, each pair of randomly selected particles will implement crisscross search on every dimension. It accelerates the global convergence of PSO without sacrificing swarm diversity [27], [28]. Inspired by CSO, a variable-scale search is proposed in this section to enhance the overall convergence ability of MEPSO. Differing from CSO, the variable-scale search only considers the impact of the best companion particle for each particle, and it does not modify the position of the best companion. Upon updating the velocity and position of each particle in the light of Eqs. (10) and (2), the variable-scale search is implemented based on distance between particle and its best companion on each dimension. The variable-scale search is defined by

$$x_{\text{tmp}}^d = x^d + r \times (\text{cbest}^d - x^d)$$

where $r$ stands for a random variable following the uniform distribution of $[-1, 1]$. In addition, a competitive mechanism is adopted to guarantee a better position. Namely, $x_{\text{tmp}}$ can survive only when it obtains a better fitness. As seen from Eq. (11), the variable-scale search is proposed to control search scale of each particle with iteration. Specifically, at the beginning of search, because the population has a rich diversity, the distance between each particle and its best companion is relatively large, and each particle will perform extensive search to obtain potential solutions with high quality. Then, with the increase of iteration, the population tends to converge, and the distance becomes relatively small. In contrast, each particle will implement local search to refine the high quality solution. Thus, each particle performs the variable-scale search from extensive range to the range around itself, which can enhance the overall convergence of MEPSO.

D. ANALYSIS OF MEPSO

1) COMPLEXITY ANALYSIS

The computational cost of the canonical PSO algorithm includes initialization, evaluation, and velocity and position update. Their complexities are $O(mn)$, $O(mn)$, and $O(2mn)$ in turn, where $m$ and $n$ represent the swarm size and the dimension of the solution space respectively. As a result, the time complexity of the canonical PSO is $O(mn)$. Similarly, initialization $O(mn)$, evaluation $O(mn)$, and velocity and position update $O(2mn)$ are also involved in MEPSO. Additionally, MEPSO needs to update the companions, calculate the influence coefficients $ggr_i$ and $cgr_i$, and perform variable-scale search. In this paper, each particle will update its companions and calculate the influence coefficients only when the personal best position is not updated within consecutive $T$ generations. In the worst-case, all particles in population reselect their companions, which causes the time complexity of $O(mn)$. Meanwhile, in such a case, we calculate the distance between each particle and its exemplars, and this step is performed for whole swarm resulting in the time complexity of $O(mn)$. Based on the distances obtained, the time complexity for calculating $ggr_i$ of whole swarm is $O(m)$, because the time
complexity for searching $\text{dist}_{g_{\text{max}}}$ in Eq. (4) is $O(m)$, and such a result can be directly used for all particles. On the other hand, as the time complexity for calculating the $\text{dist}_{c_{\max,i}}$ of each particle in Eq. (5) is $O(n)$, the total time complexity for calculating the $\text{cgr}_i$ of whole swarm is $O(mn)$. In addition, since each particle employs variable-scale search on every dimension, the time complexity of whole swarm to perform this operation is $O(mn)$. To summarize, on basis of above analysis, the final time complexity of MEPSO is also $O(mn)$, which maintains the same order of magnitude to that of the canonical PSO.

2) MOVEMENT ANALYSIS

Based on the velocity and position update equations of PSO, i.e., Eqs. (1) and (2), it is obvious that each particle moves towards the global best position independently as shown in Figure. 1 (a). In MEPSO, best companion particle as well as global best particle determines the new position of each particle. As shown in Figure. 1 (b), with the multi-exemplar learning based on influent coefficient, the lagged particle, which is far away from the global best particle, receives relatively less effect from the global best particle in comparison with the effect it received in Figure. 1 (a).

Namely, such a lagged particle prefers to carry local search, contributing to improve population diversity. In addition, each particle will perform the variable-scale search in terms of distance between particle and its best companion on each dimension, resulting in different search scales in different dimension, which is shown in Figure. 1 (c).

E. FLOW-PATH OF MEPSO

To describe MEPSO clearly, the pseudo-code of MEPSO is shown in Algorithm 1, and the flow chart of MEPSO is shown in Figure. 2.

Algorithm 1 Pseudo-Code of MEPSO

Input: $\text{MaxFEs}$: maximum function evolution number, $N$: population size, $T$: maximum number of generations for reselecting companion particles

Output: $\text{gbest}$: global best position, $\text{gbestfit}$: global best fitness

01: Initialize velocity $v_i$ and position $x_i$ of each particle $(1 \leq i \leq N)$;
02: Evaluate $\text{fitness}_i$ of each particle;
03: Initialize personal best position $pbest_i$ and personal best fitness $pbestfit_i$;
04: Initialize global best position $\text{gbest}$ and global best fitness $\text{gbestfit}$;
05: Initialize companion particles for each particle;
06: Calculate distance between each particle and its exemplars;
07: Calculate global influence coefficient $\text{ggr}_i$ and companion influence coefficients $\text{cgr}_i$;
08: for $\text{FEs} = 1 : \text{MaxFEs}$
09:   for $i = 1 : N$
10:      if $\text{flag}_i == T$
11:         Reselect companion particles;
12:         Calculate distance between $ith$ particle and its exemplars;
13:         Calculate $\text{ggr}_i$ and $\text{cgr}_i$ by Eqs. (4) and (5);
14:         $\text{flag}_i \leftarrow 0$;
15:      end if
16:      Calculate $v_i$ and $x_i$ by Eqs. (10) and (2);
17:      Evaluate $\text{fitness}_i$;
18:      Perform variable-scale search according to Eq. (11) and generate temporary position $x_{\text{tmp}}$;
19:      Evaluate $\text{fitness}_{\text{tmp}}$;
20:      if $\text{fitness}_{\text{tmp}} < \text{fitness}_i$, then
21:         $x_i \leftarrow x_{\text{tmp}}$;
22:         $\text{fitness}_i \leftarrow \text{fitness}_{\text{tmp}}$;
23:      end if
24:      if $\text{fitness}_i < pbestfit_i$, then
25:         $pbestfit_i \leftarrow \text{fitness}_i$;
26:         $pbest_i \leftarrow x_i$;
27:         $\text{flag}_i \leftarrow 0$;
28:      else
29:         $\text{flag}_i \leftarrow \text{flag}_i + 1$;
30:      end if
31:   end for
32: Update $\text{gbest}$ and $\text{gbestfit}$;
33: end for

IV. EXPERIMENT

In this section, to demonstrate the effectiveness of MEPSO, the benchmark CEC2013 test suite is adopted in our
TABLE 1. Cec2013 benchmark functions.

| No. | Functions                              | Bias  |
|-----|----------------------------------------|-------|
| 1   | Sphere Function                        | -1400 |
| 2   | Rotated High Conditioned Elliptic Function | -1300 |
| 3   | Rotated Bent Cigar Function             | -1200 |
| 4   | Rotated Discus Function                 | -1100 |
| 5   | Different Powers Function               | -1000 |

Unimodal Functions

| No. | Functions                              | Bias  |
|-----|----------------------------------------|-------|
| 6   | Rotated Rosenbrock’s Function          | -900  |
| 7   | Rotated Schaffer’s F7 Function         | -800  |
| 8   | Rotated Ackley’s Function              | -700  |
| 9   | Rotated Weierstrass Function           | -600  |
| 10  | Rotated Griewank’s Function            | -500  |
| 11  | Rastrigin’s Function                   | -400  |
| 12  | Rotated Rastrigin’s Function           | -300  |
| 13  | Non-Continuous Rotated Rastrigin’s Function | -200 |
| 14  | Schwefel’s Function                    | -100  |
| 15  | Rotated Schwefel’s Function            | 100   |
| 16  | Rotated Katunara Function              | 200   |
| 17  | Lanaeck Bi-Rastrigin Function           | 300   |
| 18  | Rotated Lanaeck Bi-Rastrigin Function  | 400   |
| 19  | Expanded Griewank’s plus Rosenbrock’s Function | 500 |
| 20  | Expanded Scaffer’s F6 Function         | 600   |

Multimodal Functions

| No. | Functions                              | Bias  |
|-----|----------------------------------------|-------|
| 21  | Composition Function 1 (n=5, Rotated)  | 700   |
| 22  | Composition Function 2 (n=3, Unrotated)| 800   |
| 23  | Composition Function 3 (n=3, Rotated)  | 900   |

Composition Functions

| No. | Functions                              | Bias  |
|-----|----------------------------------------|-------|
| 24  | Composition Function 4 (n=3, Rotated)  | 1000  |
| 25  | Composition Function 5 (n=3, Rotated)  | 1100  |
| 26  | Composition Function 6 (n=5, Rotated)  | 1200  |
| 27  | Composition Function 7 (n=5, Rotated)  | 1300  |
| 28  | Composition Function 8 (n=5, Rotated)  | 1400  |

Search range on each dimension: [-100,100]

TABLE 2. Parameter settings.

| Algorithm     | Parameters settings                  |
|---------------|--------------------------------------|
| PSO           | $w = 0.9 - 0.4$, $c_1 = c_2 = 2.05$ |
| F-PSO         | $w = 0.9 - 0.4$, $c_1 = c_2 = 2.05$ |
| DEPSO         | $w = 0.9 - 0.4$, $c_1 = c_2 = 2.05$ |
| OLPso         | $w = 0.9 - 0.4$, $c = 2.0$, $G = 5$ |
| PSOdds        | $w = 0.9 - 0.4$, $c_1 = c_2 = 2.05$ |
| CCPSO-MS      | $w = 0.6$, $c = 2.0$, $G = 5$, $P = 0.05$ |
| SRPSO         | $w = 1.05 - 0.5$, $c_1 = c_2 = 1.5$ |
| HCLPSO        | $w = 0.99 - 0.2$, $c_1 = 2.5 - 0.5$, $c_2 = 0.5 - 2.5$, $c_3 = 3 - 1.5$, $g_1 = 15$, $g_2 = 25$ |
| GLPSO         | $w = 0.7289$, $c_1 = c_2 = 1.49618$, $c_3 = 0.5$, $l = 4$, $\delta = 0.2$, $F_i \in [-1, -0.4] \cup [0.4, 1]$ |
| EPSO          | $w = 0.9 - 0.4$, $c_1 = c_2 = 1$, $c_3 = 2.0$ |
| XPSO          | $w = 0.9 - 0.4$, $c_1, c_2, c_3 \sim N(1.35, 0.1^2)$ |
| MEPSO         | $w = 0.9 - 0.4$, $c_1, c_2, c_3 \sim N(\mu_i, 0.1^2)$, $\mu = 1.0$ |

A. PARAMETER SETTING

In our experiments, the dimension $D$ of solution space and the maximum number of function evaluation $MaxFEs$ are set as 30 and 300000 respectively in the light of the stipulation of 2013 IEEE Congress on Evolutionary Computation (CEC) [29]. The population size and the maximum number of generations for reselecting companion particles $T$ are set as 50 and 5 empirically. Additionally, to determine the three acceleration coefficients, i.e., $c_1$, $c_2$ and $c_3$ in Eq.(10), the self-adapting acceleration coefficient adjustment method proposed by Xia et al. [23] is adopted in this paper to control these three coefficients. Specifically, we first initialize these three coefficients of each particle using normal distribution $N(\mu_i, 0.1^2)$, where $\mu_i$ represents the expectation of $N(\mu_i, 0.1^2)$, and 0.1 stands for the standard deviation. According to the characteristic of the normal distribution, the acceleration coefficients of each particle are perturbed around $\mu_i$. Moreover, during the search process, we adjust $\mu_i$ of each particle in terms of the experience of its current exemplars $\mu_i \leftarrow (1 - \eta) \times \mu_i + \eta \times \text{avg}(<c_i\text{exemplar}>)$, $i = 1, 2, 3$ (12)
where \( \eta \) represents the learning rate, and \( \eta = 0.1 \) has been proven to be effective in [23]. \( c_{i_{\text{exemplar}}} \) is the \( i \)th acceleration coefficient of the exemplars, and \( \text{avg}(<c_{i_{\text{exemplar}}>}) \) represents the mean acceleration coefficient of exemplars. In each generation, \( c_i \) of each particle obeys \( N(\mu_i, 0.1^2) \). Namely, \( c_i \) perturbs around the current \( \mu_i \). Thus, once the initialized \( \mu_i \) is given in prior, the current \( \mu_i \) can be updated based on Eq. (12), and \( c_i \) can be further generated with \( N(\mu_i, 0.1^2) \).
In our experiment, we initialize the same $\mu$ for all particles in swarm, i.e., $\mu_1 = \mu_2 = \mu_3 = \mu$, based on the method given by Xia et al. [23], and test different $\mu$ on all 28 functions of CEC2013 test suite. Figure 3 shows the fitness of six functions with different $\mu$.

As seen from Figure 3, on $f_6$, $f_{10}$, $f_{12}$ and $f_{28}$, MEPSO obtains its best performance with the lowest fitness when $\mu$ is 1.0. In addition, on $f_1$ and $f_{22}$, MEPSO obtains the best performance when $\mu$ is 1.4 or 1.2 respectively. However, the performance of MEPSO is relatively weak on the other functions when $\mu$ is 1.4 or 1.2. Fortunately, the performance of MEPSO with $\mu$ of 1.0 is just weaker than that of MEPSO with $\mu$ of 1.3 and 1.4 on $f_1$. Meanwhile, the performance of MEPSO with $\mu$ of 1.0 is just weaker than that of MEPSO with $\mu$ of 1.2 on $f_{22}$. Finally, we set $\mu$ as 1.0 by comprehensive considering the fitness of MEPSO with different $\mu$ on the 28 functions.

**B. EFFECTIVENESS TEST**

To demonstrate the effectiveness of Influence Coefficient (IC) and Variable-scale Search (VS) proposed in this study, we select 15 benchmark functions in CEC2013 test suite for testing. The fitness of MEPSO, MEPSO-VS, and MEPSO-IC versus Function Evolutions (FEs) on these 15 functions is shown in Figure 4. Note that, MEPSO-IC and MEPSO-VS denote the algorithms that the IC and VS are removed from MEPSO respectively.
MEPSO-IC. In addition, on \( f_5 \) and \( f_9 \), the final fitness of MEPSO is the same to that of MEPSO-IC. On \( f_{26} \), the final fitness of MEPSO is the same to that of MEPSO-VS. Namely, for most functions MEPSO performs better than MEPSO-VS and MEPSO-IC while MEPSO performs as same as MEPSO-IC or MEPSO-IC for the other functions. That is because at the beginning of search, an extensive search is performed for MEPSO, contributing to finding several solutions with high quality. With the increase of iteration, population tends to converge. In contrast, each particle will implement local search to refine the high quality solution. Moreover, MEPSO can find a better solution even at the end of the search, e.g., on \( f_{13} \) and \( f_{16} \), because IC expands the swarm diversity of MEPSO.

### C. COMPARISONS OF MEPSO WITH PSO-BASED ALGORITHMS

In this section, the canonical PSO and 10 state-of-the-art PSO variants, i.e., Frankenstein’s PSO (F-PSO) [30], OLPSO [31], DEPSO [32], PSODDS [33], CCPSO-ISM [34], SRPSO [17], HCLPSO [35], GLPSO [36], EPSO [37], and XPSO [23], are selected to compare with MEPSO in terms of the mean of fitness. The parameter settings of these algorithms are listed in Table 2, and they are set based on the values given in the original papers. To obtain the mean of fitness, each algorithm is performed 30 independent runs on each function. Table 3 illustrates the mean of fitness with respect to these 12 algorithms. The best results among these 12 algorithms on each function are marked in bold and underline type. To further illustrate the average rank of these 12 algorithms and the performance differences between MEPSO and the other 11 PSO-based algorithms, the Friedman test with significance level \( \alpha = 0.05 \) is adopted in this section, and the comparing results are listed in Table 4 and Table 5.

As seen from Table 3, for the 5 unimodal functions (\( f_1 - f_5 \)), MEPSO achieves the best performance with the lowest fitness on \( f_4 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions. The rank of MEPSO is just not as good as that of GLPSO, but it is better than those of the other 10 algorithms.

In Table 3, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.

In Table 3, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.

In Table 3, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.

In Table 4, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.

In Table 3, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.

In Table 3, \( f_6 - f_{20} \) belong to the multimodal functions of CEC2013 test suite. As shown in Table 3, MEPSO achieves the best performance with the lowest fitness on \( f_8 \), OLPSO, PSODDS, XPSO and GLPSO obtain the best performance on \( f_1, f_2, f_3 \) and \( f_5 \) respectively. Meanwhile, in terms of the Friedman test results in Table 4, we can see that MEPSO obtains the best performance with rank of 3.40 on these 5 unimodal functions, which is better than XPSO and EPSO with ranks of 4.20 and 4.60 respectively.
## TABLE 6. Comparisons of mepso with 11 pso-based algorithms in terms of real time (s).

| Algorithms | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ |
|------------|------|------|------|------|------|------|------|------|------|-------|
| MEPSO      | 10.84 | 11.12 | 12.92 | 11.65 | 10.63 | 12.16 | 16.49 | 16.62 | 36.70 | 12.96 |
| PSO        | 5.79  | 6.40  | 6.43  | 6.15  | 5.96  | 5.85  | 11.60 | 10.86 | 20.39 | 9.71  |
| F-PSO      | 12.63 | 14.11 | 14.54 | 13.65 | 12.88 | 13.65 | 17.20 | 16.49 | 37.96 | 13.93 |
| DEPSO      | 11.09 | 16.74 | 16.07 | 13.51 | 11.36 | 12.01 | 16.58 | 15.07 | 47.39 | 15.89 |
| OLPSO      | 9.75  | 9.75  | 9.70  | 9.00  | 8.82  | 9.30  | 12.97 | 12.46 | 38.13 | 10.63 |
| PSODDS     | 20.05 | 22.63 | 23.04 | 22.29 | 20.20 | 20.14 | 25.48 | 25.44 | 43.60 | 21.10 |
| CCEPSO-ISM | 12.53 | 13.72 | 13.57 | 13.38 | 11.86 | 12.74 | 18.80 | 17.60 | 42.89 | 14.66 |
| SRPSO      | 14.95 | 16.86 | 17.22 | 15.95 | 16.02 | 16.40 | 19.25 | 18.35 | 39.55 | 17.45 |
| HCLPSO     | 12.28 | 12.29 | 13.30 | 12.25 | 12.02 | 12.46 | 16.11 | 16.32 | 38.89 | 13.08 |
| GLPSO      | 15.84 | 16.21 | 16.35 | 15.73 | 15.64 | 15.64 | 18.71 | 33.75 | 16.34 | 16.72 |
| EPSO       | 14.53 | 20.03 | 18.20 | 18.07 | 14.20 | 19.75 | 21.78 | 23.15 | 42.66 | 16.40 |
| XPLO       | 12.96 | 14.80 | 14.11 | 13.78 | 13.76 | 14.40 | 16.34 | 16.11 | 35.02 | 13.56 |

| Algorithms | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ |
|------------|------|------|------|------|------|------|------|------|------|-------|
| MEPSO      | 13.73 | 14.74 | 15.88 | 14.06 | 14.50 | 18.93 | 14.24 | 18.12 | 12.65 | 15.84 |
| PSO        | 8.97  | 9.78  | 9.55  | 8.73  | 8.92  | 10.67 | 6.95  | 7.24  | 7.03  | 8.39  |
| F-PSO      | 14.62 | 15.59 | 15.39 | 14.52 | 14.71 | 17.99 | 13.77 | 14.54 | 13.35 | 14.81 |
| DEPSO      | 13.07 | 14.20 | 13.66 | 13.24 | 13.58 | 19.05 | 12.41 | 12.24 | 12.23 | 11.73 |
| OLPSO      | 11.07 | 11.95 | 11.89 | 11.14 | 11.14 | 15.39 | 10.25 | 10.93 | 9.75  | 10.54 |
| PSODDS     | 19.99 | 20.31 | 21.52 | 20.51 | 21.64 | 23.67 | 20.91 | 20.48 | 19.39 | 20.09 |
| CCEPSO-ISM | 14.34 | 16.42 | 17.06 | 15.91 | 16.37 | 20.33 | 14.41 | 14.98 | 13.65 | 15.70 |
| SRPSO      | 17.61 | 18.16 | 17.26 | 16.75 | 16.58 | 19.57 | 16.16 | 16.53 | 15.20 | 16.82 |
| HCLPSO     | 13.95 | 14.39 | 13.88 | 13.06 | 13.56 | 19.05 | 13.00 | 14.05 | 12.87 | 13.11 |
| GLPSO      | 17.47 | 17.45 | 16.84 | 19.55 | 16.30 | 19.55 | 16.30 | 16.81 | 15.79 | 16.39 |
| EPSO       | 16.65 | 19.30 | 19.27 | 21.41 | 23.35 | 28.50 | 15.82 | 16.06 | 16.33 | 22.56 |
| XPLO       | 14.87 | 15.48 | 16.51 | 15.35 | 16.67 | 19.57 | 14.93 | 17.58 | 14.46 | 16.26 |

algorithms with rank of 3.13 on these 8 composition functions. In summary, MEPSO achieves the best comprehensive performance with the rank of 3.70 on all CEC2013 functions.

As seen from Table 5, we use $P$-value and chi-square statistic on basis of Friedman test (significance level $\alpha = 0.05$) to indicate the performance differences between MEPSO and the other 11 PSO-based algorithms. On basis of the definition of $P$-value, if $P$-value is less than or equals to 0.05, such case means that it has a significant difference for each pair of compared algorithms. we can see from Table 5 that, MEPSO is better than PSO, F-PSO, DEPSO, OLPSO, PSODDS, CCEPSO-ISM and SRPSO significantly on the 28 functions based on $P$-value. In addition, the chi-square statistic is utilized to evaluate the difference between each pair of compared algorithms. Specifically, the larger the chi-square statistic is, the greater the difference between the compared algorithms. On the 28 functions, in terms of chi-square statistic, MEPSO is much different from PSO, DEPSO, SRPSO, PSODDS, F-PSO, OLPSO and CCEPSO-ISM when comparing it with the other 4 algorithms.

Moreover, in Table 6 we compare the real time of these 12 algorithms. As seen from Table 6, on $f_1$ - $f_6$ and $f_{10}$ the real time of MEPSO is just longer than those of PSO and OLPSO. However, it is faster than other 9 algorithms. For other functions, MEPSO is relatively longer. That is because the extra time consumption mainly occurs for IC calculation, and some complex functions, e.g., $f_{18}$ and $f_{25}$, need frequent IC calculation resulting extra time consumptions in comparison with simple functions. Fortunately, MEPSO achieves the best comprehensive performance when comparing the other 11 algorithms in Table 4. In addition, MEPSO almost maintains the same order of magnitude as real time of the other 11 algorithms from Table 6. Furthermore, the time complexity analysis in Section IV can also verify such a result.

### D. COMPARISONS OF MEPSO WITH NON-PSO OPTIMIZATION ALGORITHMS

As discussed in table IV, MEPSO performs better than the other 11 PSO-based algorithms in terms of comprehensive performance. In this section MEPSO is further compared with 3 non-PSO optimization algorithms, i.e., modified artificial bee colony algorithm with neighborhood search (MABC-NS) [38], evolution strategy with covariance matrix adaptation (CMAES) [39], and population-based incremental
In this paper, we propose a multi-exemplar particle swarm optimization, named MEPSO, to improve the diversity and convergence. Specifically, the global best particle and best companion of each particle are both considered as its exemplars, which brings more useful knowledge for particle update. To further describe the influences with respect to different exemplars, we define two influence coefficients inspired by mechanics. Such influence coefficients ensure that the best current experience is shared while enrich the diversity of population. Moreover, the variable-scale search is given based on the distance between particle and its best companion particles to enhance the overall convergence ability. To verify the effectiveness of our algorithm, the canonical PSO, 10 state-of-art PSO variants and 3 representative non-PSO optimization algorithms are compared with MEPSO on CEC2013 test suite. Experimental results demonstrate that MEPSO performs better than the 14 competitors in terms of comprehensive performance.

Currently, we are focusing on the following extensions to the proposed algorithm. First, to further enhance the performance of MEPSO, we try to combine MEPSO with other well performed optimization algorithms, considering their advantages and disadvantages for combination. Moreover, to address better applications of our algorithm and realize it as an effective optimizer in depth, we are now trying to develop its applications to discrete optimization and multi-objective optimization.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a multi-exemplar particle swarm optimization, named MEPSO, to improve the diversity and convergence. Specifically, the global best particle and best companion of each particle are both considered as its exemplars, which brings more useful knowledge for particle update. To further describe the influences with respect to different exemplars, we define two influence coefficients inspired by mechanics. Such influence coefficients ensure that the best current experience is shared while enrich the diversity of population. Moreover, the variable-scale search is given based on the distance between particle and its best companion particles to enhance the overall convergence ability. To verify the effectiveness of our algorithm, the canonical PSO, 10 state-of-art PSO variants and 3 representative non-PSO optimization algorithms are compared with MEPSO on CEC2013 test suite. Experimental results demonstrate that MEPSO performs better than the 14 competitors in terms of comprehensive performance.

Currently, we are focusing on the following extensions to the proposed algorithm. First, to further enhance the performance of MEPSO, we try to combine MEPSO with other well performed optimization algorithms, considering their advantages and disadvantages for combination. Moreover, to address better applications of our algorithm and realize it as an effective optimizer in depth, we are now trying to develop its applications to discrete optimization and multi-objective optimization.

TABLE 7. Comparisons of mepso with 3 non-psp optimization algorithms in terms of fitness.

|          | MEPSO | MABC-NS | CMAES | PBILc |
|----------|-------|---------|-------|-------|
| f1       | 3.41E-13 | 5.38E-13 | 4.40E-13 | 7.58E-14 |
| f2       | 2.08E+06 | 3.57E+06 | 4.47E+13 | 2.28E+06 |
| f3       | 1.53E+07 | 2.31E+08 | 8.90E+02 | 3.25E+06 |
| f4       | 1.24E+02 | 2.68E+04 | 4.40E+13 | 3.96E+04 |
| f5       | 5.42E+13 | 8.53E+13 | 7.77E+13 | 9.03E+01 |
| f6       | 5.18E+01 | 5.83E+01 | 1.76E+00 | 3.21E+01 |
| f7       | 7.18E+00 | 6.63E+01 | 1.41E+01 | 1.82E+00 |
| f8       | 2.09E+01 | 2.09E+01 | 2.15E+01 | 2.09E+01 |
| f9       | 1.69E+01 | 2.57E+01 | 4.30E+01 | 1.22E+01 |
| f10      | 6.85E-02 | 3.65E-01 | 3.36E-02 | 5.01E-03 |
| f11      | 2.06E+01 | 3.32E-02 | 9.44E+01 | 4.21E+01 |
| f12      | 4.84E+01 | 1.72E+02 | 8.69E+02 | 1.60E+02 |
| f13      | 8.36E+01 | 2.41E+02 | 1.73E+03 | 1.57E+02 |
| f14      | 9.24E+02 | 5.08E+01 | 5.95E+03 | 1.43E+03 |
| f15      | 5.35E+03 | 3.94E+03 | 5.25E+03 | 1.33E+03 |
| f16      | 2.25E+00 | 2.40E+00 | 7.30E-02 | 2.44E+00 |
| f17      | 5.61E+01 | 3.04E+01 | 4.07E+03 | 1.89E+02 |
| f18      | 1.68E+02 | 3.00E+01 | 3.99E+03 | 1.95E+02 |
| f19      | 2.56E+00 | 1.16E+00 | 3.44E+00 | 3.38E+00 |
| f20      | 1.21E+01 | 1.25E+01 | 1.28E+01 | 1.31E+01 |
| f21      | 2.95E+02 | 3.10E+02 | 3.12E+02 | 3.19E+02 |
| f22      | 7.34E+02 | 9.36E+01 | 7.24E+03 | 1.20E+03 |
| f23      | 4.50E+03 | 4.53E+03 | 6.67E+03 | 1.24E+03 |
| f24      | 2.25E+02 | 2.41E+02 | 7.44E+02 | 2.18E+02 |
| f25      | 2.73E+02 | 2.78E+02 | 3.40E+02 | 2.44E+02 |
| f26      | 2.09E+02 | 2.00E+02 | 5.22E+02 | 2.98E+02 |
| f27      | 5.96E+02 | 8.09E+02 | 5.68E+02 | 4.90E+02 |
| Rank     | 2.05   | 2.48    | 3.05   | 2.41   |

TABLE 8. Comparisons of mepso with 3 non-pso optimization algorithms in terms of p-value and chi-square.

|          | MEPSO vs |
|----------|----------|
|          | P-value  | chi-square |
| MABC-NS  | 0.072    | 3.24       |
| CMAES    | 0.050    | 3.85       |
| PBILc    | 0.239    | 1.38       |

REFERENCES

[1] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.
[2] Y.-B. Wang, D.-G. Chang, Y.-H. Fan, G.-J. Zhang, J.-Y. Zhan, X.-J. Shao, and W.-L. He, “Acoustic localization of partial discharge sources in power transformers using a particle-swarm-optimization-route-searching algorithm,” IEEE Trans. Dielectr. Electr. Insul., vol. 24, no. 6, pp. 3647–3656, Dec. 2017.
[3] F. Valdez, P. Melin, and O. Castillo, “Modular neural networks architecture optimization with a new nature inspired method using a fuzzy combination of particle swarm optimization and genetic algorithms,” Inf. Sci., vol. 270, pp. 143–153, Jun. 2014.
[4] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, “Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms,” Inf. Sci., vol. 305, pp. 77–92, Jun. 2015.
[5] Q. Lin, S. Liu, Q. Zhu, C. Tang, J. Chen, C. A. C. Coello, K.-C. Wong, and J. Zhang, “Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems,” IEEE Trans. Evol. Comput., vol. 22, no. 1, pp. 32–46, Feb. 2018.
[6] Y.-H. Jia, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, W.-J. Yu, and J. Zhang, “A dynamic logistic dispatching system with set-based particle swarm optimization,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 48, no. 9, pp. 1607–1621, Sep. 2018.
[7] W.-N. Chen, J. Zhang, H. S. H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, “A novel set-based particle swarm optimization method for discrete optimization problems,” IEEE Trans. Evol. Comput., vol. 14, no. 2, pp. 278–300, Apr. 2010.
[8] Y. Shi and R. C. Eberhart, “Parameters selections in particle swarm optimization,” in Proc. IEEE Int. Conf. Evol. Prog., Mar. 1998, pp. 591–600.
[9] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in Proc. Conge. Evol. Comput., Jul. 1999, pp. 1945–1950.
[10] A. Ratanaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 240–255, Jun. 2004.
[12] E. K. Aydogan, Y. Delice, U. Özcan, C. Gencer, and Ö. Bali, “Balancing stochastic U-lines using particle swarm optimization,” J. Intell. Manuf., vol. 30, no. 1, pp. 97–111, Jan. 2019.

[13] M. Isiet and M. Gadala, “Self-adapting control parameters in particle swarm optimization,” Appl. Soft Comput., vol. 83, pp. 105653–105677, Oct. 2019.

[14] T. Peram, K. Veeramachaneni, and C. K. Mohan, “Fitness-distance-ratio based particle swarm optimization,” in Proc. IEEE Swarm Intell. Symp., Apr. 2003, pp. 174–181.

[15] Q. Liu, W. Wei, H. Yuan, Z.-H. Zhan, and Y. Li, “Topology selection for particle swarm optimization,” Inf. Sci., vol. 363, pp. 154–173, Oct. 2016.

[16] Y. Shi and R. C. Eberhart, “Fuzzy adaptive particle swarm optimization,” in Proc. Congr. Evol. Comput., May 2001, pp. 101–106.

[17] M. R. Tanweer, S. Suresh, and N. Sundararajan, “Self-regulating particle swarm optimization algorithm,” Inf. Sci., vol. 294, pp. 182–202, Feb. 2015.

[18] C. Li, S. Yang, and T. T. Nguyen, “A self-learning particle swarm optimizer for global optimization problems,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 3, pp. 627–646, Jun. 2012.

[19] X. Jin, Y. Liang, D. Tian, and F. Zhuang, “Particle swarm optimization using dimension selection methods,” Appl. Math. Comput., vol. 219, no. 10, pp. 5185–5197, Jan. 2013.

[20] Y. Li, Z.-H. Zhan, S. Lin, J. Zhang, and X. Luo, “Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems,” Inf. Sci., vol. 293, pp. 370–382, Feb. 2015.

[21] N. Lynn and P. N. Suganthan, “Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation,” Swarm Evol. Comput., vol. 24, pp. 11–24, Oct. 2015.

[22] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, “Genetic learning particle swarm optimization,” IEEE Trans. Cybern., vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[23] N. Lynn and P. N. Suganthan, “Ensemble particle swarm optimizer,” Appl. Soft Comput., vol. 55, pp. 533–548, Jun. 2017.

[24] X. Zhou, H. Wang, M. Wang, and J. Wan, “Enhancing the modified artificial bee colony algorithm with neighborhood search,” Soft Comput., vol. 21, no. 10, pp. 2733–2743, May 2017.

[25] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” Evol. Comput., vol. 9, no. 2, pp. 159–195, Jun. 2001.

[26] A. Ducoulombier and A. Ducoulombier, “Extending population-based incremental learning to continuous search spaces,” in Proc. Int. Conf. Parallel Problem Solving Nature, Amsterdam, The Netherlands, 1998, pp. 418–427.