A Particle Swarm Optimization with Adaptive Learning Weights Tuned by A Multiple-Input Multiple-Output Fuzzy Logic Controller

Xuewen Xia, *Member, IEEE,* Haojie Song, Yinglong Zhang, Ling Gui, Xing Xu, Kangshun Li, Yuanxiang Li

***Abstract*—In a canonical particle swarm optimization (PSO) algorithm, the fitness is a widely accepted criterion when selecting exemplars for a particle, which exhibits promising performance in simple unimodal functions. To improve a PSO's performance on complicated multimodal functions, various selection strategies based on the fitness value are introduced in PSO community. However, the inherent defects of the fitness-based selections still remain. In this paper, a novelty of a particle is treated as an additional criterion when choosing exemplars for a particle. In each generation, a few of elites and mavericks who have better fitness and novelty values are selected, and saved in two archives, respectively. Hence, in each generation, a particle randomly selects its own learning exemplars from the two archives, respectively. To strengthen a particle's adaptive capability, a multiple-input multiple-output fuzzy logic controller is used to adjust two parameters of the particle, i.e., an acceleration coefficient and a selection proportion of elites. The experimental results and comparisons between our new proposed PSO, named as MFCPSO in this paper, and other 6 PSO variants on CEC2017 test suite with 4 different dimension cases suggest that MFCPSO exhibits very promising characteristics on different types of functions, especially on large scale complicated functions. Furthermore, the effectiveness and efficiency of the fuzzy controlled parameters are discussed based on extensive experiments.**

***Index Terms*—Particle Swarm Optimization, Adaptive Learning Weights, Fuzzy Logic Controller, Elites and Mavericks, Global Optimization.**

I. INTRODUCTION

ALONG with the rapid improvement of the computer's calculation ability, optimization is playing an increasingly important role in various fields, such as engineering, data mining, and scientific problems [1]–[4]. Although some traditional analytical methods still offer favorable performance on some simple optimization tasks, they do not display promising characteristics on those multimodal, large scale or noisy problems.

To deal with such problems, some evolutionary algorithms (EAs) have attracted much more attention in the last decades due to their reliable and comprehensive performance. Particle swarm optimization (PSO) algorithm, as a popular EA, was proposed by Kennedy and Eberhart in 1995 [5], [6]. During the optimization process, each particle in PSO adjusts its search direction and step relying on helpful knowledge extracted from different exemplars. Although a single particle in PSO has very low intelligence, collective behaviors cause a population to display a powerful capability in optimizing various problems [3], [7], [8].

However, similar to other EAs, the canonical PSO also suffers from the premature convergence when optimizing complicated problems though it can achieve a higher convergence speed. It is generally accepted by the PSO community that maintaining an appropriate population diversity is beneficial for preventing the premature convergence and improving the exploration ability [9]. However, it is not a practical and feasible approach that mere seeking for the population diversity because it is harmful for the exploitation ability of PSO. Thus, considering that many real applications are "black-box" problems, some researchers have poured attention into achieving favorable balance between the exploration and the exploitation by proper parameters adjustments [10]–[14] and learning models [15]–[17].

Generally, an effective learning model of a particle relies on its exemplars and corresponding learning weights. In the canonical PSO [5], [6], a learner particle selects its own historical best position and the global best particle, measured by fitness values, as its exemplars to perform the learning process. The simple fitness-based selection of exemplars enables the canonical PSO to offer very promising and efficient performance on simple unimodal functions due to that the outstanding exemplars can help the learner particle find out the global optima with a high convergence speed. However, the selection method exclusively considering the fitness may cause the population to be easily trapped into local optima when optimizing some complicated multimodal functions. Hence, to overcome the inherit weaknesses of the fitness-based selection, some study incorporate various disturbances during the search process, which can be deemed a randomness-based selection, intending to bring a high population diversity, and then improve the exploration capability [18], [19].

Unlike pursuing a solution with the best fitness in many EAs, searching for a system without an explicit objective has captured some researchers' attention in artificial life fields [20]. A widely accepted method in the fields is generating a complex artificial system with a more novelty rather than a higher fitness. A few studies verify that the novelty-based

Manuscript received ****; revised ****.

X. Xia, H. Song, Y. Zhang, L. Gui, and X. Xu are with the College of Physics and Information Engineering, Minnan Normal University, China; and the Key Lab of Intelligent Optimization and Information Processing, Minnan Normal University, Zhangzhou, 363000, China (e-mail: xwxia@whu.edu.cn, 970162167@qq.com, laughkid@163.com, xxustar@qq.com, zhang_yinglong@126.com).

K. Li is with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China (e-mail: likangshun@sina.com).

Y. Li is with the School of Computer, Wuhan University, Wuhan, China (e-mail: yxli@whu.edu.cn).

search is immune to problems of deception and local optima, which are inherent in the fitness-based search, since the novelty-based search entirely ignores an explicit objective. Some results support a counter-intuitive conclusion that disregarding (or partial disregarding) the objective in this way may be beneficial for searching for the objective [21]–[23]. However, it is dangerous to regard that the novelty-based search dominates the traditional fitness-based search. In fact, the novelty-based search sacrifices performance of the exploitation capability though it is favorable for the exploration ability. In other words, the fitness-based search and the novelty-based search have their own merits. Thus, rationally and efficiently utilizing the two search mechanisms may bring very comprehensive performance for PSO.

Based on the above discussions, this paper proposes a PSO variant based on a ***M***ultiple-input multiple-output (MIMO) ***F***uzzy logic ***C***ontroller, named as MFCPSO in this paper. In each generation, two archives are used to save a few selected elites and mavericks, who have better fitness and higher novelty values, respectively. During the search process, each particle randomly select two particles respectively selected from the two archives as its exemplars. Moreover, to efficiently utilizing the two exemplars, two accelerate coefficients which can be regarded as two learning weights of the two exemplar, and a selection ratio are adjusted by the MIMO fuzzy logic controller. The main characteristics of MFCPSO can be summarized as follows.

1) Instead of using fitness as a single criterion for exemplars selection, the novelty is considered as an additional criterion.
2) In each generation, a particle randomly selects two exemplars from the two types of candidate exemplars who respectively have better fitness and higher novelty. Thus, the particle has two exemplars with distinct properties.
3) During the search process, weights of a particle learning to the two exemplars are controlled by a fuzzy logic controller, aiming to take advantages of distinct merits of the exemplars. As a result, various requirements of different search stages can be satisfied.

The rest of this paper is organized as follows. Section II presents a framework of the canonical PSO and reviews some PSO studies. Details of MFCPSO are described in Section III. The experimental results and corresponding discussions are detailed in Section IV. Finally, conclusions are presented in Section V.

II. Related Works

A. Canonical PSO

In PSO, a particle i at generation t is associated with two vectors, i.e., a position vector $\mathbf{X}_i^t = [x_{i,1}^t, x_{i,2}^t, ..., x_{i,D}^t]$ and a velocity vector $\mathbf{V}_i^t = [v_{i,1}^t, v_{i,2}^t, ..., v_{i,D}^t]$, where D represents the dimension of a problem under study. The vector $\mathbf{X}_i^t$ is regarded as a candidate solution while the vector $\mathbf{V}_i^t$ is treated as a search direction and step size of the particle i at generation t. During the search process, the particle adjusts its flight trajectory based on two vectors, named as personal historical best position $\mathbf{PB}_i^t = [pb_{i,1}^t, pb_{i,2}^t, ..., pb_{i,D}^t]$ and its neighbors' best-so-far position $\mathbf{NB}_i^t = [nb_{i,1}^t, nb_{i,2}^t, ..., nb_{i,D}^t]$, respectively. The update rules of $\mathbf{V}_i^t$ and $\mathbf{X}_i^t$ are defined as Eq. (1) and Eq. (2), respectively.

$$v_{i,j}^t = w \cdot v_{i,j}^{t-1} + c_1 \cdot r_{1,j} \cdot (pb_{i,j}^t - x_{i,j}^{t-1}) + c_2 \cdot r_{2,j} \cdot (nb_{i,j}^t - x_{i,j}^{t-1}) \quad (1)$$

$$x_{i,j}^t = x_{i,j}^{t-1} + v_{i,j}^t \quad (2)$$

where w represents an inertia weight determining how much the information of previous search is preserved; c_1 and c_2 are two acceleration coefficients deciding relative learning weights for $\mathbf{PB}_i^t$ and $\mathbf{NB}_i^t$, respectively; $r_{1,j}$ and $r_{2,j}$ are two random numbers uniformly distributed in the interval [0, 1]; $x_{i,j}^t$ and $v_{i,j}^t$ represent the jth dimension values of $\mathbf{X}_i^t$ and $\mathbf{V}_i^t$, respectively. Note that, when a particle i regards all other particles as its neighbors, $\mathbf{NB}_i^t$ is the historical global best position of the entire population.

B. Study of PSO

In PSO community, designing an efficient velocity update rule has captured many researchers' attention. It can be observed from Eq. (1) that three parameters (i.e., w, c_1, and c_2) and two learning exemplars (i.e., $\mathbf{PB}_i^t$ and $\mathbf{NB}_i^t$) play crucial roles in improving the exploration and the exploitation abilities. Thus, majority of PSO variants proposed in the last decades are focused on parameters adjustments and learning exemplars selections, which are briefly reviewed hereinafter.

1) Parameters Adjustments: There's a general consensus in the PSO community that a population should pay more attention on the exploration and the exploitation in the early search stage and the later search stage, respectively. Thus, various time-varying parameters are proposed in last decades. For instance, the most ubiquitous update rule of w is linearly decreasing from 0.9 to 0.4 over the entire search stage which is still applied in many PSOs now [10]. Motivated by the study, Ratnaweera *et al.* [24] further advocated a PSO with time-varying acceleration coefficients, i.e., c_1 and c_2, in HPSO-TVAC. The experimental results manifest that a larger c_1 is beneficial for keeping the diversity of population at the early search stage, while a larger c_2 is propitious to speed up the convergence at the later search stage. However, considering that the search process of PSO is nonlinear and complicated, various nonlinear-varying strategies are proposed to tune the parameters [11]–[13], [25] aiming to give particles diverse search behaviors.

To layout a more flexible and satisfactory adjustment for the crucial parameters, various adaptive adjustments taking advantages of historical information of the population have been proposed in last few years [26], [27]. For instance, adjustments of w, c_1, and c_2 in [27] no longer rely on (or only rely on) iteration numbers. Instead, particles' fitness [14], [28]–[31] and velocity [32] are selected as criteria when adjusting the parameters. Extensive experimental results verify that the adaptive strategies can exhibit a proper trade-off between the exploration and the exploitation, and then endow PSO with more comprehensive and reliable performance.

This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

2) Learning Exemplars Selections: In the canonical PSO, the global version PSO (GPSO) and the local version PSO (LPSO) are two basic topological structures when a particle choosing its learning exemplars [15]. Generally, it is a common strategy that a particle selects its own historical best position and its neighbors historical best position as learning exemplars. However, the exemplars selections cannot efficiently deal with a deception problem lied in complicated multimodal tasks. To overcome the shortcoming, many researchers adopt comprehensive information of multiple particles to generate exemplars for a particle. In this sense, the comprehensive learning strategy [34], the orthogonal learning strategy [35], the interactive learning strategy [36], the dimensional learning strategy [37], and the multiple exemplars [31] are remarkable works.

Motivated by the division of labour in human society, Li *et al*. [38] proposed a self-learning PSO (SLPSO) in which particles are assigned four different roles according to distinct local fitness landscapes that the particles belong to. Accordingly, the different roles representing four distinct learning strategies enable the particles to independently deal with various situations. Moreover, Xia also proposed a fitness-based multi-role PSO (FMPSO) [29], in which different particles in a subpopulation play different roles, and then select their own learning exemplars to perform distinct search behaviors. Extensive experiments in these studies demonstrate that it is a promising strategy to satisfy distinct requirements in different search stages that assigning different roles to different particles according to their properties.

III. MFCPSO

A. Motivations of MFCPSO

In the majority of PSO variants, it is a widely accepted strategy that evaluating a particle with respect to the specific objective function. Based on the measuring results, only those particles with higher fitness have more chance to be exemplars. Although the fitness-based exemplars selection mechanism is intuitively reasonable, it may cause a population to be easily trapped into local optima when optimizing complicated multimodal problems. In recent years, on the contrary, many studies in artificial life often focuses on tasks without explicit objectives. For instance, a common approach is to create an open-ended system by searching for good behavioral novelties instead of pursuing high fitness values [21], [22]. A few studies on neural networks [39] and robotics systems [21], [23] also verify that the novelty-based search process can offer very promising properties on many tasks.

Why using the novelty instead of the fitness as a driving force for individuals? The motivation is that a deceptive fitness landscape in a complicated problem may cause a search algorithm following the fitness gradient to perform worse. On the contrary, ignoring the objective fitness entirely (or partially), the algorithm cannot be (or not easily to be) deceived with respect to the objective. In the following, a maze experiment based on PSO is used to illustrate the performance of the fitness-based driving and the novelty-based driving.

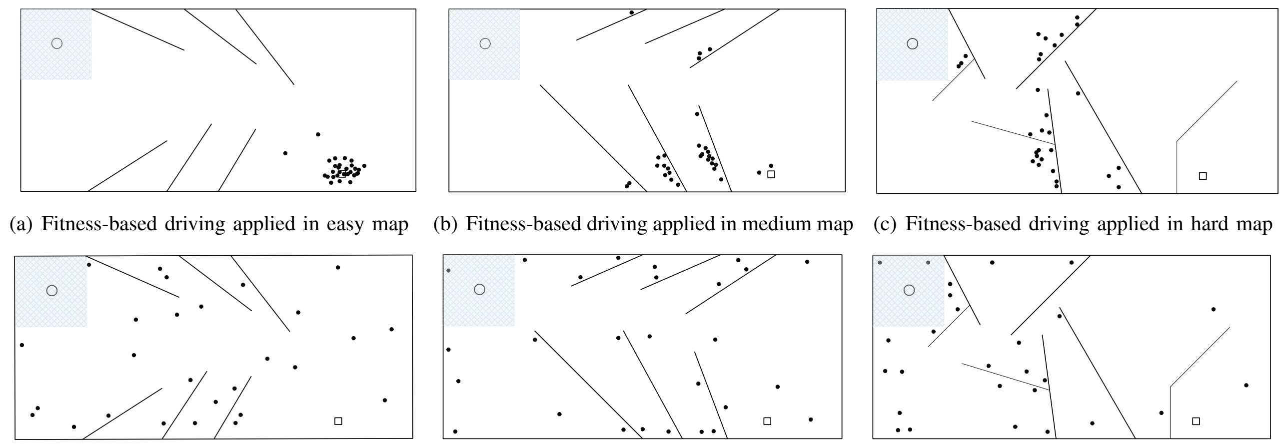
In the experiment, 3 maze maps with different difficulties are introduced. Details of the 3 maps, respectively noted as easy map, medium map, and hard map. To illustrate comprehensive characteristics of the fitness-based driving force and the novelty-based driving force in different circumstances, 6 experiments are performed. Concretely, in each experiment, a population with 30 particles is randomly generated in a shadow rectangle located in the upper left corner of the map. Then, each particle selects its own learning exemplars based on fitness or novelty to perform the search process. After 1000 generations, the population stops its search process. Final positions of all particles over 6 typical runs on different maps are demonstrated by Fig. 1.

From results of Fig. 1(a) and Fig. 1(d) we can see that the fitness-based driving force can help all particles convergent to the goal position. On the contrary, no particle can search the goal position under the novelty-based driving force, though the population has a higher diversity. On the medium map, who has many simple traps, only 2 particles in the population can reach the goal position under the fitness-based driving force, while a few particles under the novelty-based driving force can reach the area round the goal position. Moreover, the population under the novelty-based driving force still keeps higher diversity. On the hard map, many difficult traps cause the population to fall into the traps, and all particles are far away from the goal. On the contrary, a few particles driven by the novelty reach the goal region. It can be observed from the comparison results that the fitness-based driving force yields more favorable characteristics than the novelty-based driving force in simple problems who have no traps. On the contrary, the novelty-based driving force dominates the fitness-based driving force on complicated problems who have many difficult traps.

Thus, inspired by the improvements in artificial life and the comparison results introduced above, we think that the novelty value as well as the fitness value also can be regarded as a criterion when choosing exemplars for a specific particle. As a result, the particle can extract different knowledge from two different types of exemplars which are separately selected based on the two criteria. Concretely, an exemplar with higher novelty can enhance the particle's exploration ability, while an exemplar with better fitness can improve the particle's exploitation ability. Thus, we regard that adjusting the learning weights from exemplars who have higher novelty or fitness values is a promising strategy to satisfy various requirements of the exploration and exploitation in different optimization stages.

During the last decades, fuzzy logic theory has attracted many scholars' attentions due to its superior uncertainty and noise-handling ability from the usage of human-like linguistic variables [40], [41]. Furthermore, the fuzzy logic theory also has exhibited distinctive and outstanding characteristics on control field [42]–[44] and optimization field [45], [46].

Thus, in this study, we will apply a MIMO fuzzy logic controller to tune a particle's learning weights of the two different types of exemplars who have higher novelty or fitness values. Relying on the fuzzy controlled weights, particles in different search stages pour different attentions on fitness or novelty (i.e., the exploitation or the exploration), and then satisfy distinct requirements of different search stages.



(a) Fitness-based driving applied in easy map (b) Fitness-based driving applied in medium map (c) Fitness-based driving applied in hard map

(d) Novelty-based driving applied in easy map (e) Novelty-based driving applied in medium map (f) Novelty-based driving applied in hard map

Fig. 1: Final positions of all particles over typical runs. Each maze map depicts a typical run, stopping at 1000 generations, and each small black point represents a particle.

To implement aforementioned motivations, three main steps are involved in MFCPSO. Firstly, particles with better fitness value (i.e., elites) or higher novelty value (i.e., mavericks) should be separately saved in two archives. Secondly, each particle randomly chooses its two exemplars from the two archives in each generation. Lastly, a fuzzy logic system is used to adjust each particle's parameters, and then control its learning weights on the two exemplars. Detailed information of MFCPSO is introduced as follows.

B. Saving Elites and Mavericks

In this study, two exemplars of a particle i in the canonical PSO, i.e., $\mathbf{PB}_i$ and $\mathbf{NB}_i$, are replaced by two distinct exemplars, i.e., an elite and a maverick who have a promising fitness value and a high novelty value, respectively. Due to that the fitness associated with a function value is a widely known term in EA field, the definition of it is not discussed in this part.

Since a maverick is measured by its "novelty" value, thus, how to define a particle's novelty must be firstly dealt with. Generally, a novelty value denotes how unique a particle's behavior is. In this study, for simplicity, a particle's novelty is calculated by an average distance between the particle and its the K-nearest neighbors. As a result, the novelty of the particle $\mathbf{X}_i$ can be defined as Eq. (3).

$$novel(\mathbf{X}_i^t) = \frac{1}{K}\sum_{j=1}^{K} dist(\mathbf{X}_i, \mu_j) \tag{3}$$

where μ_j is the jth-nearest neighbor of $\mathbf{X}_i$; and $dist(\mathbf{X}_i, \mu_j)$ denotes Euclidean distance between $\mathbf{X}_i$ and μ_j.

In each generation, the personal best positions (i.e., $\mathbf{PB}_i$) of all particles are sorted according to their fitness, and the $p \cdot N$ best results are saved in an archive $\mathcal{A}_{\mathcal{E}}$, where N denotes the population size. Note that, a solution's fitness $fit(\mathbf{X})$ denotes an error value $f(\mathbf{X}) - f(\mathbf{X}^*)$, where $f(\mathbf{X})$ is the function value of the solution $\mathbf{X}$, and $\mathbf{X}^*$ denotes the real global optimum of a problem. Meanwhile, the current positions $\mathbf{X}_i$ of all the particles are also sorted according to their novelty measured by Eq. (3), and then the $(1-p) \cdot N$ best positions, in terms of the novelty value, are saved in another archive $\mathcal{A}_{\mathcal{M}}$.

During the search process, the update of the two archives can be described as **Algorithm** 1. Without loss of generality, minimization problems are considered in this paper. In other words, the lower fitness value $fit(\mathbf{X}_i)$ is, the better performance $\mathbf{X}_i$ has. Considering that not only different particles demonstrate distinct properties in a generation, but also a same particle may offer diverse characteristics in different generations, we assign distinct p_i to each particle i aiming to satisfy distinct requirements of different search processes.

Algorithm 1. Update_Archives ( )

Input: $\mathbf{PB}_i^t$, $\mathbf{X}_i^t$, and p_i^t;
1: $\mathcal{A}_{\mathcal{E}} = \mathcal{A}_{\mathcal{M}} = \emptyset$; $Ne = \lceil p_i * N \rceil$; $Nm = \lceil (1-p_i) * N \rceil$;
2: Sort $\mathbf{PB}_i^t$ in ascending order, i.e.
$fit(\mathbf{PB}_{i_1}^t) \le fit(\mathbf{PB}_{i_2}^t) \le ... \le fit(\mathbf{PB}_{i_N}^t)$;
3: $\mathcal{A}_{\mathcal{E}}$={$\mathbf{PB}_{i_1}^t$, $\mathbf{PB}_{i_2}^t$, ..., $\mathbf{PB}_{i_{Ne}}^t$};
4: Sort $\mathbf{X}_i$ in descending order, i.e.
$novel(\mathbf{X}_{i_1}^t) \ge novel(\mathbf{X}_{i_2}^t) \ge ... \ge novel(\mathbf{X}_{i_N}^t)$;
5: $\mathcal{A}_{\mathcal{M}}$={$\mathbf{X}_{i_1}^t$, $\mathbf{X}_{i_2}^t$, ..., $\mathbf{X}_{i_{Nm}}^t$};
Output: $\mathcal{A}_{\mathcal{E}}$, $\mathcal{A}_{\mathcal{M}}$.

Note that, $\lceil p * N \rceil$ in **Algorithm** 1 denotes that rounding up the value $p * N$ to an integer.

C. Update of Velocity

In MFCPSO, the update of velocity is based on the two archives, i.e., $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$. In each generation, the particle i randomly selects two exemplars, named as $\mathbf{E}_i$ and $\mathbf{M}_i$, from $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$, respectively. According to the definition of $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$ we can see that the particle i can obtain different knowledge from the two exemplars. Because different particles may display distinct properties and need to execute distinct search behaviors, thus, each particle should have different learning weights on the two exemplars. As a result, in this study, a particle i has its own two acceleration coefficients, i.e.,

$c_{i,1}$ and $c_{i,2}$. Accordingly, the particle i updates its velocity based on Eq.(4).

$$v_{i,j}^t = w \cdot v_{i,j}^{t-1} + c_{i,1} \cdot r_{1,j} \cdot (e_{i,j}^t - x_{i,j}^{t-1}) + c_{i,2} \cdot r_{2,j} \cdot (m_{i,j}^t - x_{i,j}^{t-1}) \quad (4)$$

where $e_{i,j}$ and $m_{i,j}$ are the j^{th} dimension values of $\mathbf{E}_i$ and $\mathbf{M}_i$, respectively; $c_{i,1}$ and $c_{i,2}$ are two acceleration coefficients of the particle i.

we can observe From Eq. (4) that $c_{i,1}$ and $c_{i,2}$ decide the learning weights that the particle i learns knowledge from exemplars $\mathbf{E}_i$ and $\mathbf{M}_i$, respectively. Considering that $\mathbf{E}_i$ and $\mathbf{M}_i$ respectively have a higher fitness value and a greater novelty value, we regard that a larger $c_{i,1}$ and a smaller $c_{i,2}$ are beneficial for learning much helpful information from elites and extracting little knowledge from mavericks. As a result, the exploitation capability of the particle i can be improved. On the contrary, a smaller $c_{i,1}$ and a larger $c_{i,2}$ can help the particle i learn much more knowledge from $\mathbf{M}_i$ rather than $\mathbf{E}_i$. Thus, the exploration ability of the particle i can be enhanced. To satisfy distinct requirements of a fitness landscape that a particle belongs to, the particle should be able to adjust its own acceleration coefficients in different search stages. In this study, $c_{i,1}$ and $c_{i,2}$ are controlled by a fuzzy logic system. Details of the control process are introduced in Section III-D.

Furthermore, it can be see From **Algorithm** 1 that the parameter p determines the size of the two archives. Concretely, a greater p makes $\mathcal{A}_{\mathcal{E}}$ have a larger size. In this condition, the archive $\mathcal{A}_{\mathcal{E}}$ can save more $\mathbf{PB}_i$ with a high fitness value. Thus, a $\mathbf{PB}_i$ saved in the archive with a relatively lower fitness also has a chance to be selected as an exemplar $\mathbf{E}_i$ for a particle i. Meanwhile, a small size $\mathcal{A}_{\mathcal{M}}$ caused by the greater p only saves a small amount of $\mathbf{X}_i$ with a very higher novelty value. As a result, an exemplar $\mathbf{M}_i$ randomly selected from $\mathbf{X}_i$ can provide more novelty information for the particle i. The above discussions denote that a greater p can give the particle i more chance to learn from exemplars with relatively lower fitness, as well as bring more novelty information to the particle. In other words, a greater p is beneficial for the exploration ability. On the contrary, a smaller p enables that only a few $\mathbf{PB}_i$ with a very high fitness value can be candidate exemplars for the particle i. Thus, we regard that a smaller p is favorable for the exploitation ability.

Due to a fact that the particle i may display distinct properties in different search processes, it needs to adjust its p accordingly. The control process of p_i based on a fuzzy logic system is detailed in the following section.

D. Parameters Controlled by Fuzzy Logic

In recent years, various fuzzy logic control methods have been successful applied in many EAs, and offer distinct positive characteristic [46]–[48]. Thus, in MFCPSO, a MIMO fuzzy logic system is responsible for determining $c_{i,1}$, $c_{i,2}$, and p_i of the particle i, the values of which control the learning weights for two types of exemplars (i.e., elites and mavericks). The fuzzy control system makes decisions based on three types of information about the particle i.

1) Ideal Control Objective: When utilizing a fuzzy controller to control a system, an ideal control objective is essential. Considering that the search process of a population is a dynamic progress, the ideal control objective of the fuzzy controller should be adjusted during the search process. In this study, two adjusted ideal control objectives, named as ideal fitness Fit_{ideal} and ideal novelty Nov_{ideal}, need to be set during the optimization process. Concretely, in each generation, the average fitness and average novelty are regarded as Fit_{ideal} and Nov_{ideal}, respectively. Thus, the definitions of the two ideal control objectives at the generation t can be defined as Eq. (5) and Eq. (6), respectively.

$$Fit_{ideal}^t = \frac{1}{N} \cdot \Sigma_{i=1}^N fit(\mathbf{X}_i^t) \quad (5)$$

$$Nov_{ideal}^t = \frac{1}{N} \cdot \Sigma_{i=1}^N novel(\mathbf{X}_i^t) \quad (6)$$

2) Fuzzy Control of Parameters: In this study, $c_{i,1}$, $c_{i,2}$, and p_i of each particle i are controlled by a MIMO fuzzy logic controller. In each generation t, $efit_i^t$ and $enov_i^t$ of the particle i are two inputs of the controller, which are defined as Eq. (7) and Eq. (8), respectively.

$$efit_i^t = Fit_{ideal}^t - fit(\mathbf{X}_i^t) \quad (7)$$

$$enov_i^t = novel(\mathbf{X}_i^t) - Nov_{ideal}^t \quad (8)$$

After the definitions of the state variables input to the MIMO fuzzy controller, the two crisp input values (i.e., $efit_i^t$ and $enov_i^t$) are transformed into fuzzy values via the process of fuzzification.

Commonly, there may be significant difference in function values between two different functions. Moreover, from Eq. (5) we can observe that the value of Fit_{ideal} changes dynamically due to that the value of $fit(\mathbf{X}_i^t)$ may be different in each generation. Thus, it is unfeasible to predefine a fixed range for $efit_i^t$. In this work, the maximum value and the minimum value of $efit_i^t$, denoted as $max(efit_i^t)$ and $min(efit_i^t)$ respectively, are determined in each generation. Based on the upper and lower limitation values in a generation, each $efit_i^t$ can be transformed into a fuzzy value as follows.

For each input, we have defined 7 fuzzy sets which denote negative big (NB), negative medium (NM), negative small (NB), zero (ZO), positive small (PS), positive medium (PM), and positive big (PB), respectively. Before the fuzzification process, $max(efit_i^t)$ and $min(efit_i^t)$ need to be calculated. Then, the range [$max(efit_i^t)$, $min(efit_i^t)$] is divided into 6 equal parts: Δ=($max(x)$-$min(x)$)/6. Based on the value of Δ, thus, the membership function employed to $efit_i$ can be seen in Fig. 2(a). Note that the fuzzification process of $enov_i^t$ is similar as that of $efit_i^t$. Thus, it's not described in this part.

In each generation t, after the inputs are converted from real values into fuzzy values, the MIMO logic controller makes determinations for c_i^t and p_i^t based on two sets of the fuzzy rules as shown in Table I and Table II, respectively. To illustrate the fuzzy inference process based on Table I and Table II, two fuzzy rules marked with the star symbol are given as follows:

Implication #1:

IF $efit_i^t$ is NB and $enov_i^t$ is NB,

TABLE I: Set of fuzzy rules of $efit_i^t$ and $enov_i^t$ for $c_{i,1}^t$.

$c_{i,1}^t$ \ $efit_i^t$ / $enov_i^t$	NB	NM	NS	ZO	PS	PM	PB
NB	NB$^{*(\#1)}$	NM	NS	ZO	ZO	ZO	ZO
NM	NB	NM	NS	ZO	ZO	ZO	PS
NS	NM	NS	ZO	ZO	ZO	ZO	PS
ZO	NM	NS	ZO	ZO	ZO	PS	PM
PS	NS	ZO	ZO	ZO	ZO	PS	PM
PM	NS	ZO	ZO	ZO	PS	PM	PB
PB	ZO	ZO	ZO	ZO	PS	PM	PB$^{*(\#2)}$

THEN $c_{i,1}^t$ is NB and p_i^t is PB.

Implication #2

IF $efit_i^t$ is PB and $enov_i^t$ is PB,

THEN $c_{i,1}^t$ is PB and p_i^t is NB.

In *Implication* #1, the particle *i* displays the most unfavorable performance measured by fitness value, while it has the lowest novelty. In this case, we can regard that the particle *i* may be trapped into a local optimum. Thus, a greater $c_{i,1}^t$ helps the particle learn much more knowledge from elites, which is beneficial for increasing fitness. Meanwhile, a greater p_i^t enables the particle to select two exemplars from large-scale $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$. As a result, the particle can jump out of the local optimum and then pay more attention on searching other promising regions. On the contrary, in *Implication* #2, the particle *i* offers the most favorable performance measured by both fitness and novelty. In such a situation, on the one hand, the particle needs a greater $c_{1,i}^t$ to increase its learning weight for other elites, intending to further improve its fitness. On the other hand, the particle also can use a larger p_i^t to find out whether there are better solutions in other regions.

After the fuzzy inference process, the obtained results, i.e., $c_{i,1}^t$ and p_i^t, are fuzzy values located in a range of output values. To obtain two crisp values of the results, the process of defuzzification, which is the exact opposite of fuzzification process, needs to be conducted. Note that, membership functions of p_i^t and $c_{i,1}^t$ are demonstrated in Fig. 2(b) and Fig. 2(c), respectively. The applied defuzzification strategy in our MIMO controller is the centriod of area, which allows to define the output as Eq. (9).

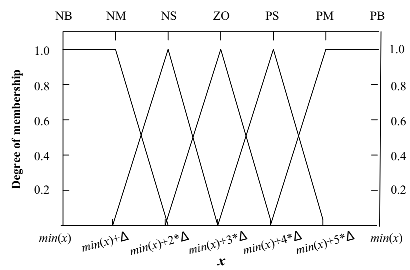
$$f = \frac{\Sigma_{k=1}^{|R|} y^k \cdot \mu_{B^k}(y^k)}{\Sigma_{k=1}^{|R|} \mu_{B^k}(y^k)} \tag{9}$$

where f is the defuzzified output result; $|R|$ is the number of rules of the fuzzy system; $\mu_{B^k}(y^k)$ and y^k are a membership function and the output result of kth rule, respectively.

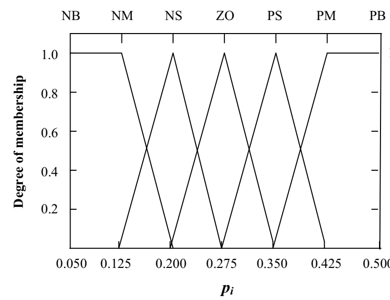
Finally, two output variables, i.e., $c_{i,1}^t$ and p_i^t, of the MIMO logic controller are fed back to the PSO algorithm module, and then each particle's learning weights for elites and mavericks are adjusted.

E. Framework of MFCPSO

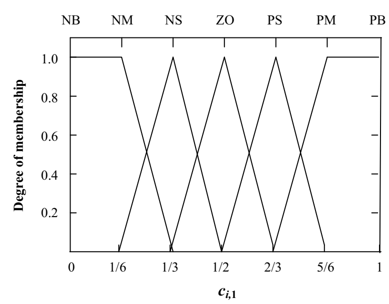
By incorporating the aforementioned components, the general idea of MFCPSO is demonstrated by Fig. 3, and the pseudo code of it is shown in **Algorithm 2**.



(a) Membership function of $efit_i$ and $enov_i$



(b) Membership function of p_i



(c) Membership function of $c_{1,i}$

Fig. 2: The graphical illustration of membership functions applied in MFCPSO. Two inputs, i.e. $efit_i$ and $enov_i$, use a same membership function described as Fig. 2(a), in which x denotes $efit_i$ (or $enov_i$) and Δ=($max(x)$-$min(x)$)/6. Two membership functions of the two outputs, i.e. p_i and $c_{i,1}$, are illustrated by Fig. 2(b) and Fig. 2(c), respectively. Note that, value ranges of p_i and $c_{i,1}$ are [0.05, 0.50] and [0.0, 1.0], respectively.

TABLE II: Set of fuzzy rules of $efit_i^t$ and $enov_i^t$ for p_i^t.

p_i^t \ $efit_i^t$ / $enov_i^t$	NB	NM	NS	ZO	PS	PM	PB
NB	PB$^{*(\#1)}$	PM	PS	ZO	ZO	ZO	ZO
NB	PB	PM	PS	ZO	ZO	ZO	NS
NS	PM	PS	ZO	ZO	ZO	ZO	NS
ZO	PM	PS	ZO	ZO	ZO	NS	NM
PS	PS	ZO	ZO	ZO	ZO	NS	NM
PM	PS	ZO	ZO	ZO	NS	NM	NB
PB	ZO	ZO	ZO	ZO	NS	NM	NB$^{*(\#2)}$

IV. Experimental Studies

A. Benchmark Functions and Peer Algorithms

In this work, the CEC2017 test suite is utilized to testify the performance of MFCPSO. In the test suite, 30 benchmark functions are categorized into 4 different types, i.e., unimodal functions (F_1-F_3), simple multimoal functions (F_4-F_{10}), hybrid functions (F_{11}-F_{20}), and composition functions functions

This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

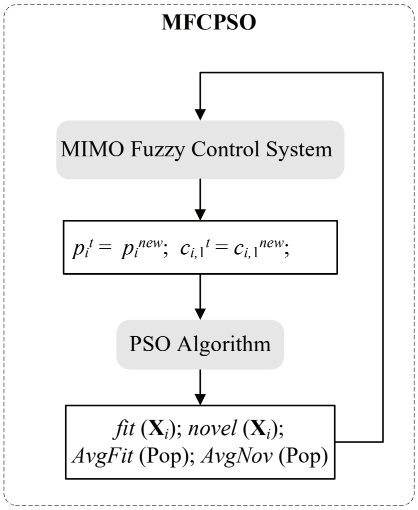


Fig. 3: The general idea of MFCPSO.

Algorithm 2. MFCPSO

/* Initialization */
Initialize N, $t = 1$, $\mathcal{A}_{\mathcal{E}} = \mathcal{A}_{\mathcal{M}} = \emptyset$, p_i^t, $c_{i,1}^t$, $c_{i,2}^t$, and $K = \lceil N/20 \rceil$;
For i=1 to N **Do**
  Randomly initialize $\mathbf{V}_i^t$, and $\mathbf{X}_i^t$; Evaluate $\mathbf{X}_i^t$; $\mathbf{PB}_i^t = \mathbf{X}_i^t$;
End For
Update **GB**;
/* Main Loop */
While not meeting terminal conditions
  For i=1 to N **Do**
    Update $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$ based on **Algorithm 1**;
    Randomly select E_i and M_i from $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{M}}$, respectively;
    Update the velocity and position by Eqs. (4) and (2), respectively;
    Evaluate $\mathbf{X}_i^t$, and Update $\mathbf{PB}_i^t$;
  End For;
  Update **GB**;
  Compute Fit_{ideal} and Nov_{ideal} by Eqs. (5) and (6), respectively;
  For i=1 to N **Do**
    Compute $efit_i^t$ and $enov_i^t$ by Eqs. (7) and (8), respectively;
    Update p_i^t and $c_{i,1}^t$ by the fuzzy controller (see Section III-D);
    $c_{i,2}^t = 1 - c_{i,1}^t$;
  End For
  $t = t + 1$;
End While
Output result **GB**.

(F_{21}-F_{30}). Detailed information of the functions can refer to the literature [49]. To explore the scalability of MFCPSO, four different dimension cases of the CEC2017 test suite, i.e., *D*=10, 30, 50, and 100, are tested, while the maximum number of function evaluations ($MaxFEs$) is set to $10000 \times D$ for each dimension case.

Note that, other 6 state-of-the-art PSO variants, including CCPSO-ISM [50], SRPSO [14], GLPSO [51], XPSO [31], TAPSO [52] and AWPSO [4], are chosen as peer algorithms. Parameters settings of all the peer algorithms are summarized in Table III.

To obtain statistical results, each peer algorithm is carried out 51 independent runs on each function. The experimental results, measured by mean value (*Mean*) and standard deviation (*S. D.*), on the four dimension cases are detailed in Table IV - Table VII, respectively. The best results of the *Mean* on each function among all algorithms are marked with shadow background. Moreover, results of *t*-test between MFCPSO and the other 6 competitors as well as rank values (the lower the better) of all the peer algorithms are also presented in the tables. Concretely, the results of the t-test recorded as symbols "+", "-", and "=" denote that MFCPSO is significantly better than, significantly worse than, and almost the same as a competitor algorithm, respectively. Symbols (#)+ and (#)- denote the number of "+" and "-" in each column, respectively; and *Avg.*(*Rank*) means an average value of that an algorithm attains the *Rank* values on all the test functions. Note that, freedom at a 0.05 level of significance is adopted in the t-test. For instance, the result"2.44+02(1-)" in the first row of Table IV means that the mean fitness of CCPSO-ISM on F_1 is 2.44+02, the *Rank* value of it is 1, and MFCPSO is significantly worse than CCPSO-ISM on F_1.

TABLE III: Basic Information of 9 Peer Algorithms

Algorithm	Published Year	Population size
MFCPSO	–	2*D
CCPSO-ISM [50]	2015	20/40/60/100
SRPSO [14]	2015	20/40/60/100
GLPSO [51]	2016	20/50/100/200
XPSO [31]	2020	30/50/100/200
TAPSO [52]	2020	20/40/60/100
AWPSO [4]	2020	20/40/60/100

B. Solutions Accuracy

In this part, the comparison results are presented, in terms of solutions accuracy. To provide a detailed analysis of the performance of MFCPSO, the experimental results of all the peer algorithms on the 4 different types of functions are discussed respectively.

1) Unimodal Functions (F_1-F_3): From Table IV we can see that MFCPSO and GLPSO achieve the the same performance on F_2 and F_3, measured by the solution accuracy. Furthermore, the two algorithms also display almost the same performance on F_1, in terms of the *t*-test result. Moreover, CCPSO-ISM also offers very outstanding performance on this type of functions. However, with the increasing of problem scale, the results presented in Table V - Table VII show that GLPSO exhibits more favorable characteristics than other competitors. Specially, GLPSO attains the best mean values on 2 out of the 3 simple unimodal functions in 30D, and 100D. On the contrary, MFCPSO achieves the best result on one function in 30D, and 50D, while it cannot yield the best result on any functions in 100D. Although CCPSO-ISM displays the best performance in 10D, its performance rapidly deteriorate. The results indicate that the scalability of CCPSO-ISM and MFCPSO on the unimodal functions is unfavorable.

2) Simple Multimodal Functions (F_4-F_{10}): It can be observed from the results that MFCPSO offers the best results on 6 out of the 7 simple multimodal functions on all the 4 dimension cases, followed by TAPSO who attains the most favorable performance on F_4 on all the cases. Although GLPSO exhibits unfavorable performance on the 7 simple multimodal functions when dimension of them is *D*=10, it displays very promising characteristics on the type of functions in higher dimension cases because the its *Rank* values are 2 or 3 on all the multimodal functions on the higher dimension cases. The results verify that MFCPSO as well as GLPSO has

TABLE IV: Comparison results of solution accuracy on CEC2017 test suite(D=10).

		CCPSO-ISM	SRPSO	GLPSO	XPSO	TAPSO	AWPSO	MFCPSO
F_1	*Mean*	2.44E+02 (1-)	1.38E+06(6=)	2.27E+03(4=)	2.05E+03(2=)	2.44E+03(5=)	1.11E+08(7+)	2.25E+03(3)
	S.D.	2.68E+02	9.81E+06	2.66E+03	2.11E+03	2.75E+03	3.76E+08	2.75E+03
F_2	*Mean*	0.00E+00 (1=)	1.04E+05(6=)	0.00E+00 (1=)	0.00E+00 (1=)	0.00E+00 (1=)	2.48E+05(7+)	0.00E+00 (1)
	S.D.	0.00E+00	4.78E+05	0.00E+00	0.00E+00	0.00E+00	8.30E+05	0.00E+00
F_3	*Mean*	2.31E+03(7+)	6.35E-14(5+)	0.00E+00 (1=)	2.45E-14(3+)	4.46E-14(4+)	3.20E+01(6+)	0.00E+00 (1)
	S.D.	1.19E+03	3.53E-14	0.00E+00	2.84E-14	4.73E-14	2.28E+02	0.00E+00
F_4	*Mean*	1.27E+00(4=)	5.90E+00(6+)	3.41E-02(2-)	3.54E-01(3-)	4.02E-05 (1-)	1.71E+01(7+)	1.73E+00(5)
	S.D.	1.91E+00	3.95E+00	5.05E-03	2.62E-01	1.24E-04	3.47E+01	1.50E+00
F_5	*Mean*	1.63E+01(7+)	9.18E+00(3+)	1.12E+01(5+)	7.85E+00(2+)	1.38E+01(6+)	1.06E+01(4+)	5.15E+00 (1)
	S.D.	4.92E+00	3.78E+00	4.36E+00	4.05E+00	5.70E+00	4.52E+00	2.31E+00
F_6	*Mean*	4.81E-06(2=)	5.53E-02(6=)	6.05E-06(3=)	9.94E-06(4=)	4.92E-03(5+)	2.60E-01(7+)	4.08E-06 (1)
	S.D.	1.13E-05	2.78E-01	2.66E-05	4.15E-05	1.04E-02	5.28E-01	2.13E-05
F_7	*Mean*	2.07E+01(5+)	1.80E+01(2+)	2.15E+01(7+)	1.81E+01(3+)	2.10E+01(6+)	1.85E+01(4+)	1.35E+01 (1)
	S.D.	2.74E+00	4.35E+00	5.66E+00	2.53E+00	3.98E+00	4.15E+00	2.05E+00
F_8	*Mean*	1.70E+01(7+)	7.36E+00(2+)	1.15E+01(5+)	7.75E+00(4+)	1.32E+01(6+)	7.49E+00(3+)	5.28E+00 (1)
	S.D.	4.17E+00	2.84E+00	5.45E+00	3.40E+00	4.51E+00	3.62E+00	2.58E+00
F_9	*Mean*	6.20E+00(7+)	1.27E-13(4+)	2.85E-02(5+)	0.00E+00(1=)	5.69E-01(6+)	9.14E-14(3+)	0.00E+00 (1)
	S.D.	9.55E+00	8.40E-14	1.08E-01	0.00E+00	1.20E+00	6.02E-14	0.00E+00
F_{10}	*Mean*	5.33E+02(7+)	3.38E+02(3=)	4.56E+02(5+)	3.17E+02(2=)	5.02E+02(6+)	3.85E+02(4+)	2.92E+02 (1)
	S.D.	1.36E+02	1.68E+02	2.64E+02	1.77E+02	2.31E+02	1.77E+02	1.50E+02
F_{11}	*Mean*	6.43E+00(4+)	6.46E+00(5+)	6.29E+00(3+)	3.67E+00(2+)	1.20E+01(6+)	1.94E+01(7+)	2.56E+00 (1)
	S.D.	2.39E+00	3.26E+00	4.46E+00	2.63E+00	5.76E+00	3.32E+01	2.16E+00
F_{12}	*Mean*	1.99E+04(6+)	1.61E+04(5=)	9.66E+03 (1=)	1.24E+04(4=)	1.10E+04(2=)	7.55E+05(7+)	1.20E+04(3)
	S.D.	1.93E+04	1.43E+04	8.20E+03	1.14E+04	9.30E+03	2.30E+06	1.03E+04
F_{13}	*Mean*	2.18E+02 (1-)	5.31E+03(4=)	7.83E+03(7+)	5.71E+03(5=)	3.36E+03(2=)	6.34E+03(6=)	4.39E+03(3)
	S.D.	2.92E+02	4.74E+03	6.36E+03	4.71E+03	4.65E+03	6.94E+03	3.06E+03
F_{14}	*Mean*	1.22E+02(5+)	4.32E+01(2=)	8.20E+02(6+)	3.98E+01 (1-)	1.72E+03(7+)	4.91E+01(3=)	5.03E+01(4)
	S.D.	1.40E+02	1.82E+01	7.78E+02	1.59E+01	3.85E+03	3.24E+01	3.00E+01
F_{15}	*Mean*	8.12E+01(3-)	3.52E+01 (1-)	1.02E+03(6+)	5.20E+01(2-)	1.89E+03(7+)	9.72E+01(4=)	1.78E+02(5)
	S.D.	1.31E+02	2.46E+01	1.23E+03	1.11E+02	2.79E+03	2.66E+02	2.82E+02
F_{16}	*Mean*	8.50E+01 (1-)	1.96E+02(6=)	9.06E+01(2-)	1.32E+02(4-)	1.50E+02(5-)	1.11E+02(3-)	2.30E+02(7)
	S.D.	7.08E+01	1.26E+02	1.13E+02	1.41E+02	1.25E+02	9.65E+01	1.24E+02
F_{17}	*Mean*	1.48E+01 (1-)	3.74E+01(6=)	2.78E+01(2=)	2.90E+01(3=)	4.69E+01(7=)	3.09E+01(4=)	3.67E+01(5)
	S.D.	9.87E+00	1.86E+01	2.79E+01	2.19E+01	5.05E+01	1.85E+01	1.78E+01
F_{18}	*Mean*	1.44E+03 (1-)	1.30E+04(6=)	4.24E+03(2-)	8.37E+03(4-)	7.07E+03(3-)	1.57E+04(7=)	1.22E+04(5)
	S.D.	1.27E+03	1.33E+04	5.20E+03	6.22E+03	6.76E+03	1.60E+04	1.08E+04
F_{19}	*Mean*	6.76E+01(2-)	6.62E+02(4=)	4.12E+03(6+)	3.54E+01 (1-)	5.74E+03(7+)	1.24E+03(5=)	4.81E+02(3)
	S.D.	1.29E+02	4.34E+03	3.57E+03	4.53E+01	7.58E+03	6.08E+03	1.27E+03
F_{20}	*Mean*	1.22E+01(3-)	4.73E+01(4-)	6.56E+00 (1-)	5.57E+01(5-)	8.96E+00(2-)	5.81E+01(6-)	9.98E+01(7)
	S.D.	9.55E+00	5.02E+01	9.03E+00	5.80E+01	7.89E+00	6.61E+01	5.17E+01
F_{21}	*Mean*	1.21E+02 (1-)	1.78E+02(2=)	2.01E+02(5=)	1.98E+02(4=)	2.08E+02(7+)	2.04E+02(6=)	1.94E+02(3)
	S.D.	3.64E+01	5.09E+01	3.76E+01	3.29E+01	3.29E+01	3.46E+01	3.45E+01
F_{22}	*Mean*	9.70E+01 (1=)	1.01E+02(3=)	9.92E+01(2=)	1.02E+02(5=)	1.33E+02(7=)	1.27E+02(6=)	1.01E+02(4)
	S.D.	1.98E+01	2.22E+01	1.25E+01	8.43E-01	1.63E+02	1.02E+02	6.77E+00
F_{23}	*Mean*	3.21E+02(5=)	3.12E+02(3=)	3.13E+02(4=)	3.10E+02 (1=)	3.22E+02(6+)	3.23E+02(7+)	3.11E+02(2)
	S.D.	4.64E+01	3.08E+01	6.08E+00	4.66E+00	7.98E+00	1.07E+01	5.60E+00
F_{24}	*Mean*	1.32E+02 (1-)	3.27E+02(5+)	3.23E+02(4+)	2.84E+02(3=)	3.28E+02(6+)	3.53E+02(7+)	2.70E+02(2)
	S.D.	5.18E+01	6.82E+01	6.59E+01	9.93E+01	7.65E+01	3.65E+01	1.08E+02
F_{25}	*Mean*	2.94E+02 (1-)	4.25E+02(5+)	4.33E+02(6+)	4.24E+02(4=)	4.04E+02(2=)	4.36E+02(7+)	4.15E+02(3)
	S.D.	1.20E+02	2.65E+01	2.11E+01	2.33E+01	9.22E+01	2.63E+01	2.23E+01
F_{26}	*Mean*	1.80E+02 (1-)	3.42E+02(5=)	3.35E+02(4=)	2.85E+02(2=)	4.56E+02(7+)	4.49E+02(6+)	3.05E+02(3)
	S.D.	1.07E+02	2.34E+02	2.05E+02	1.44E+02	3.38E+02	2.69E+02	1.96E+02
F_{27}	*Mean*	3.91E+02 (1=)	4.13E+02(6+)	3.97E+02(3=)	3.97E+02(2=)	3.99E+02(4=)	4.18E+02(7+)	4.00E+02(5)
	S.D.	5.04E+01	2.16E+01	1.38E+01	1.04E+01	1.75E+01	2.24E+01	1.34E+01
F_{28}	*Mean*	3.12E+02 (1-)	4.99E+02(2=)	5.26E+02(5=)	5.29E+02(6=)	5.01E+02(3=)	5.52E+02(7=)	5.17E+02(4)
	S.D.	1.29E+02	1.22E+02	1.27E+02	1.36E+02	1.38E+02	9.97E+01	1.15E+02
F_{29}	*Mean*	3.11E+02(7+)	2.73E+02(2=)	2.83E+02(5=)	2.76E+02(3=)	2.98E+02(6+)	2.82E+02(4=)	2.72E+02 (1)
	S.D.	3.45E+01	3.40E+01	2.96E+01	3.41E+01	3.95E+01	4.44E+01	3.99E+01
F_{30}	*Mean*	2.45E+04 (1-)	3.44E+05(6=)	2.26E+05(4=)	2.96E+05(5=)	1.72E+05(2=)	3.77E+05(7=)	2.16E+05(3)
	S.D.	2.85E+04	5.65E+05	3.68E+05	6.34E+05	3.11E+05	5.77E+05	4.17E+05
(#)	+	10	10	11	5	16	17	-
(#)	-	14	2	4	7	4	2	-
Avg	*(Rank)*	3.17	4.17	3.87	3.03	4.80	5.60	2.97

very promising and comprehensive performance on the simple multimodal functions.

3) Hybrid Functions (F_{11}-F_{20}): In a hybrid function, different subcomponents of variables have different properties, which causes the function is difficult to be optimized. The comparison results on the 10 hybrid functions in the 4 dimension cases show that CCPSO-ISM attains the most outstanding performance in 10D and 30D cases. On the contrary, MFCPSO displays the most favorable properties in 50D and 100D cases. Meanwhile, TAPSO also offers very more promising performance in higher dimension cases (i.e., 50D and 100D) than in lower dimension cases (i.e., 10D and 30D). The experimental results indicate that MFCPSO and

This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

TABLE V: Comparison results of solution accuracy on CEC2017 test suite(D=30).

		CCPSO-ISM	SRPSO	GLPSO	XPSO	TAPSO	AWPSO	MFCPSO
F_1	*Mean*	5.46E+03(5+)	9.42E+07(6+)	3.28E+03(3=)	4.12E+03(4=)	3.25E+03(2=)	2.03E+09(7+)	2.95E+03 (1)
	Std.Dev.	2.08E+03	3.30E+08	4.21E+03	4.34E+03	3.71E+03	2.05E+09	2.96E+03
F_2	*Mean*	7.62E+11(4+)	5.00E+29(6=)	1.52E+01 (1-)	3.61E+12(5=)	5.19E+11(3=)	1.05E+34(7=)	1.97E+10(2)
	Std.Dev.	1.02E+12	3.56E+30	2.57E+01	1.37E+13	3.20E+12	7.52E+34	6.27E+11
F_3	*Mean*	4.93E+04(7+)	4.08E-04(2-)	2.69E-06 (1-)	2.18E-03(3-)	5.92E+01(4-)	2.33E+03(6+)	4.29E+02(5)
	Std.Dev.	7.88E+03	7.83E-04	1.28E-05	7.16E-03	4.23E+02	4.75E+03	3.19E+02
F_4	*Mean*	7.60E+01(3-)	1.14E+02(5+)	3.69E+01(2-)	1.18E+02(6+)	3.12E+01 (1-)	2.09E+02(7+)	8.74E+01(4)
	Std.Dev.	1.87E+01	7.57E+01	3.07E+01	2.72E+01	3.10E+01	1.69E+02	4.04E+00
F_5	*Mean*	1.34E+02(7+)	5.14E+01(4+)	4.29E+01(2+)	4.46E+01(3+)	5.73E+01(5+)	8.00E+01(6+)	1.74E+01 (1)
	Std.Dev.	1.98E+01	1.27E+01	1.25E+01	1.31E+01	1.43E+01	2.23E+01	5.41E+00
F_6	*Mean*	1.33E-01(5+)	5.41E-01(6+)	6.03E-05(2+)	2.71E-02(4+)	1.31E-02(3+)	2.06E+00(7+)	3.60E-07 (1)
	Std.Dev.	2.38E-01	8.96E-01	1.13E-04	5.10E-02	1.35E-02	2.26E+00	6.86E-07
F_7	*Mean*	1.04E+02(6+)	7.51E+01(2+)	7.56E+01(3+)	8.28E+01(4+)	8.63E+01(5+)	1.09E+02(7+)	5.12E+01 ()
	Std.Dev.	1.12E+01	1.53E+01	1.08E+01	1.33E+01	1.60E+01	2.98E+01	1.03E+01
F_8	*Mean*	1.35E+02(7+)	5.28E+01(4+)	4.66E+01(3+)	4.54E+01(2+)	5.54E+01(5+)	7.08E+01(6+)	1.74E+01 (1)
	Std.Dev.	1.73E+01	1.45E+01	1.31E+01	1.21E+01	1.46E+01	1.94E+01	3.85E+00
F_9	*Mean*	5.77E+03(7+)	5.18E+00(3+)	3.27E+00(2+)	6.79E+00(4+)	1.13E+02(5+)	2.37E+02(6+)	0.00E+00 (1)
	Std.Dev.	1.21E+03	1.14E+01	4.21E+00	6.19E+00	1.33E+02	2.75E+02	0.00E+00
F_{10}	*Mean*	2.79E+03(5+)	2.45E+03(2+)	2.53E+03(3+)	2.98E+03(6+)	2.54E+03(4+)	3.03E+03(7+)	1.56E+03 (1)
	Std.Dev.	3.40E+02	5.84E+02	5.52E+02	5.53E+02	5.17E+02	6.82E+02	4.51E+02
F_{11}	*Mean*	1.51E+02(6+)	1.03E+02(5+)	4.22E+01(2+)	8.93E+01(3+)	9.49E+01(4+)	1.84E+02(7+)	2.83E+01 (1)
	Std.Dev.	3.39E+01	4.56E+01	2.75E+01	4.29E+01	3.49E+01	6.49E+01	2.45E+01
F_{12}	*Mean*	6.81E+05(5+)	1.03E+06(6+)	2.55E+04(2-)	1.22E+05(3=)	2.31E+04 (1-)	4.91E+07(7+)	1.39E+05(4)
	Std.Dev.	3.31E+05	2.92E+06	1.28E+04	2.49E+05	1.58E+04	9.89E+07	9.67E+04
F_{13}	*Mean*	1.97E+03 (1-)	1.59E+06(6=)	1.27E+04(3=)	1.19E+04(2=)	1.32E+04(4=)	9.12E+06(7+)	1.83E+04(5)
	Std.Dev.	1.22E+03	1.00E+07	1.26E+04	1.42E+04	1.60E+04	2.31E+07	1.88E+04
F_{14}	*Mean*	3.11E+04(7+)	1.09E+04(5=)	3.49E+03 (1=)	6.18E+03(4=)	5.41E+03(3=)	2.94E+04(6+)	3.87E+03(2)
	Std.Dev.	2.49E+04	2.70E+04	4.48E+03	5.15E+03	1.77E+04	5.21E+04	6.82E+03
F_{15}	*Mean*	2.47E+02 (-)	6.84E+03(6=)	4.82E+03(4=)	4.73E+03(2=)	5.08E+03(5=)	2.67E+04(7+)	4.76E+03(3)
	Std.Dev.	8.58E+01	7.33E+03	5.75E+03	5.47E+03	7.20E+03	3.13E+04	5.35E+03
F_{16}	*Mean*	6.58E+02(4+)	5.36E+02(2+)	7.25E+02(5+)	5.94E+02(3+)	8.12E+02(7+)	8.02E+02(6+)	3.96E+02 (1)
	Std.Dev.	1.34E+02	1.42E+02	2.88E+02	2.37E+02	2.54E+02	2.90E+02	1.51E+02
F_{17}	*Mean*	2.41E+02(4+)	2.34E+02(3+)	2.42E+02(5+)	1.71E+02(2+)	2.89E+02(7+)	2.83E+02(6+)	1.32E+02 (1)
	Std.Dev.	8.46E+01	1.14E+02	1.47E+02	8.84E+01	1.68E+02	1.51E+02	7.57E+01
F_{18}	*Mean*	1.34E+05(5+)	1.69E+05(6+)	6.89E+04(2-)	1.09E+05(4=)	3.71E+04 (1-)	2.93E+05(7+)	9.61E+04(3)
	Std.Dev.	5.90E+04	1.97E+05	5.21E+04	8.18E+04	2.86E+04	3.68E+05	5.16E+04
F_{19}	*Mean*	8.13E+01 (1-)	3.42E+04(6=)	7.00E+03(5+)	5.77E+03(3=)	5.88E+03(4=)	4.08E+05(7+)	4.03E+03(2)
	Std.Dev.	3.16E+01	1.42E+05	9.24E+03	7.35E+03	7.27E+03	1.71E+06	3.74E+03
F_{20}	*Mean*	3.40E+02(6+)	2.05E+02(3=)	2.47E+02(4+)	1.93E+02 (1=)	3.63E+02(7+)	2.50E+02(5+)	1.93E+02(2)
	Std.Dev.	8.69E+01	5.92E+01	1.42E+02	7.58E+01	1.72E+02	9.72E+01	4.23E+01
F_{21}	*Mean*	2.74E+02(6+)	2.55E+02(4+)	2.45E+02(2+)	2.48E+02(3+)	2.63E+02(5+)	2.79E+02(7+)	2.27E+02 (1)
	Std.Dev.	1.02E+02	1.42E+01	1.29E+01	1.43E+01	1.75E+01	2.08E+01	8.94E+00
F_{22}	*Mean*	8.00E+02(6+)	4.82E+02(3+)	1.70E+02(2=)	5.39E+02(4+)	7.08E+02(5+)	1.87E+03(7+)	1.00E+02 (1)
	Std.Dev.	1.32E+03	8.79E+02	4.97E+02	1.14E+03	1.21E+03	1.45E+03	0.00E+00
F_{23}	*Mean*	4.54E+02(5+)	4.64E+02(6+)	3.95E+02(2+)	4.00E+02(3+)	4.39E+02(4+)	5.45E+02(7+)	3.72E+02 (1)
	Std.Dev.	4.91E+01	4.44E+01	1.25E+01	2.14E+01	2.41E+01	7.36E+01	1.07E+01
F_{24}	*Mean*	5.53E+02(5+)	5.65E+02(6+)	4.69E+02(2+)	4.78E+02(3+)	5.16E+02(4+)	6.26E+02(7+)	4.37E+02 (1)
	Std.Dev.	1.98E+02	6.59E+01	1.57E+01	3.91E+01	2.68E+01	6.10E+01	6.86E+00
F_{25}	*Mean*	3.89E+02(2+)	3.93E+02(5+)	3.89E+02(3=)	3.98E+02(6+)	3.90E+02(4=)	4.34E+02(7+)	3.87E+02 (1)
	Std.Dev.	1.58E+00	1.31E+01	1.10E+01	1.19E+01	9.70E+00	5.27E+01	1.61E-01
F_{26}	*Mean*	5.76E+02(2+)	1.34E+03(5+)	1.16E+03(4+)	7.44E+02(3+)	1.56E+03(6+)	2.18E+03(7+)	3.71E+02 (1)
	Std.Dev.	4.05E+02	6.27E+02	5.95E+02	6.03E+02	1.02E+03	7.73E+02	3.13E+02
F_{27}	*Mean*	5.18E+02(3+)	5.48E+02(6+)	5.18E+02(2+)	5.28E+02(5+)	5.18E+02(4+)	5.74E+02(7+)	5.03E+02 (1)
	Std.Dev.	4.01E+00	3.66E+01	9.14E+00	1.52E+01	1.30E+01	4.69E+01	1.03E+01
F_{28}	*Mean*	4.51E+02(6+)	4.35E+02(5+)	3.31E+02 (1-)	3.83E+02(3-)	3.40E+02(2-)	5.99E+02(7+)	4.09E+02(4)
	Std.Dev.	1.19E+01	4.47E+01	5.10E+01	6.21E+01	5.81E+01	1.77E+02	1.99E+01
F_{29}	*Mean*	7.36E+02(6+)	6.34E+02(4+)	6.12E+02(3+)	6.08E+02(2+)	7.31E+02(5+)	7.70E+02(7+)	4.91E+02 (1)
	Std.Dev.	7.78E+01	1.08E+02	1.60E+02	9.09E+01	1.69E+02	2.12E+02	4.35E+01
F_{30}	*Mean*	1.19E+04(6+)	8.98E+03(4+)	4.85E+03 (1-)	9.21E+03(5+)	5.85E+03(2=)	5.73E+05(7+)	7.16E+03(3)
	Std.Dev.	5.23E+03	1.29E+04	2.31E+03	4.22E+03	4.26E+04	1.61E+06	3.77E+03
(#)	+	26	22	17	19	17	29	-
(#)	-	4	1	7	2	5	0	-
Avg	*(Rank)*	4.77	4.53	2.57	3.50	4.03	6.70	1.90

TAPSO dominate other competitors on this type of functions in higher dimension cases, while CCPSO-ISM is more suitable for the hybrid functions with lower dimension.

4) Composition Functions (F_{21}-F_{30}): The composition function is another type of complicated functions, in which different functions are basic components of the function. Similar as the results on the hybrid functions, the experimental results on the 10 composition functions also indicate that CCPSO-ISM attains the best results on 8 out of the 10 composition function in the 10D case. However, CCPSO-ISM cannot yield the best results on any functions in the 3 higher dimension cases. On the contrary, MFCPSO displays the most favorable characteristics on 8, 9, and 7 out of the 10 composition func-

This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

TABLE VI: Comparison results of solution accuracy on CEC2017 test suite(D=50).

		CCPSO-ISM	SRPSO	GLPSO	XPSO	TAPSO	AWPSO	MFCPSO
F_1	*Mean*	3.06E+04(5+)	6.72E+08(6+)	4.86E+03(4+)	3.70E+03(3=)	3.50E+03(2=)	7.84E+09(7+)	2.40E+03 (1)
	S.D.	2.07E+04	1.53E+09	6.11E+03	5.33E+03	6.19E+03	5.31E+09	3.56E+03
F_2	*Mean*	4.38E+26(5+)	2.33E+50(6=)	2.30E+02 (4=)	5.06E+25(3=)	2.30E+19(2=)	9.61E+62(7=)	4.62E+25(3)
	S.D.	8.19E+26	1.66E+51	5.26E+02	3.60E+26	1.64E+20	6.83E+63	2.64E+26
F_3	*Mean*	1.40E+05(7+)	2.07E+02 (1-)	4.79E+02(3-)	1.72E+03(4-)	2.18E+02(2-)	1.71E+04(5=)	1.84E+04(6)
	S.D.	1.64E+04	1.17E+03	7.26E+02	6.55E+02	6.12E+02	1.50E+04	4.55E+03
F_4	*Mean*	1.00E+02(3-)	1.71E+02(5+)	8.13E+01(2-)	2.27E+02(6+)	5.10E+01 (1-)	7.49E+02(7+)	1.14E+02(4)
	S.D.	2.16E+01	9.22E+01	4.37E+01	5.26E+01	4.59E+01	8.45E+02	2.31E+01
F_5	*Mean*	2.90E+02(7+)	1.15E+02(4+)	8.63E+01(3+)	8.51E+01(2+)	1.28E+02(5+)	1.65E+02(6+)	3.85E+01 (1)
	S.D.	2.95E+01	3.31E+01	1.90E+01	1.94E+01	2.62E+01	2.90E+01	1.33E+01
F_6	*Mean*	1.16E+00(6+)	1.13E+00(5+)	2.03E-04(2+)	3.87E-01(4+)	2.10E-02(3+)	5.99E+00(7+)	9.53E-07 (1)
	S.D.	1.30E+00	1.25E+00	3.17E-04	5.88E-01	1.71E-02	3.61E+00	1.84E-06
F_7	*Mean*	2.21E+02(6+)	1.50E+02(3+)	1.32E+02(2+)	1.72E+02(5+)	1.72E+02(4+)	2.47E+02(7+)	1.05E+02 (1)
	S.D.	2.52E+01	2.40E+01	1.58E+01	2.96E+01	2.89E+01	7.97E+01	2.36E+01
F_8	*Mean*	2.77E+02(7+)	1.11E+02(4+)	8.48E+01(2+)	8.83E+01(3+)	1.18E+02(5+)	1.63E+02(6+)	3.98E+01 (1)
	S.D.	3.11E+01	2.26E+01	2.02E+01	2.25E+01	2.13E+01	2.98E+01	1.64E+01
F_9	*Mean*	2.04E+04(7+)	1.22E+02(4+)	1.22E+01(2+)	6.91E+01(3+)	7.65E+02(5+)	2.18E+03(6+)	4.24E-14 (1)
	S.D.	3.43E+03	1.79E+02	9.61E+00	7.23E+01	7.24E+02	1.28E+03	6.81E-14
F_{10}	*Mean*	5.19E+03(6+)	4.56E+03(4+)	4.48E+03(3+)	5.12E+03(5+)	4.39E+03(2+)	5.89E+03(7+)	2.40E+03 (1)
	S.D.	4.04E+02	7.28E+02	7.84E+02	9.90E+02	6.71E+02	9.71E+02	4.86E+02
F_{11}	*Mean*	2.56E+02(6+)	1.78E+02(5+)	6.97E+01(2+)	1.67E+02(4+)	1.32E+02(3+)	4.35E+02(7+)	4.88E+01 (1)
	S.D.	5.75E+01	4.95E+01	2.49E+01	4.66E+01	3.74E+01	2.84E+02	9.92E+00
F_{12}	*Mean*	4.97E+06(5+)	3.26E+08(6+)	3.66E+05(2-)	8.39E+05(3-)	1.02E+05 (1-)	2.22E+09(7+)	1.46E+06(4)
	S.D.	1.49E+06	6.87E+08	1.82E+05	1.33E+06	1.68E+05	3.13E+09	6.10E+05
F_{13}	*Mean*	6.81E+03(5+)	7.06E+07(6+)	3.30E+03(2=)	4.41E+03(4=)	4.27E+03(3=)	3.51E+08(7+)	2.59E+03 (1)
	S.D.	7.09E+03	2.42E+08	4.28E+03	5.24E+03	5.83E+03	8.36E+08	5.48E+03
F_{14}	*Mean*	3.54E+05(7+)	7.56E+04(5+)	1.75E+04(2-)	3.92E+04(4=)	1.71E+04 (1=)	3.48E+05(6+)	2.92E+04(3)
	S.D.	1.63E+05	8.62E+04	1.15E+04	3.34E+04	3.98E+04	8.13E+05	1.74E+04
F_{15}	*Mean*	1.04E+03 (1-)	1.05E+04(6+)	4.97E+03(4=)	4.61E+03(3=)	9.15E+03(5+)	1.44E+06(7+)	4.46E+03(2)
	S.D.	4.50E+02	8.27E+03	4.83E+03	3.87E+03	6.66E+03	9.97E+06	4.64E+03
F_{16}	*Mean*	1.22E+03(5+)	7.71E+02(2+)	1.21E+03(4+)	1.08E+03(3+)	1.28E+03(6+)	1.64E+03(7+)	4.15E+02 (1)
	S.D.	2.25E+02	2.67E+02	4.23E+02	3.29E+02	4.72E+02	4.76E+02	8.55E+01
F_{17}	*Mean*	8.73E+02(5+)	8.05E+02(3+)	8.72E+02(4+)	7.87E+02(2+)	1.01E+03(6+)	1.20E+03(7+)	4.71E+02 (1)
	S.D.	1.47E+02	2.43E+02	3.04E+02	2.47E+02	2.55E+02	3.08E+02	1.38E+02
F_{18}	*Mean*	6.37E+05(6=)	2.54E+05(3-)	1.74E+05(2-)	3.73E+05(4=)	3.53E+04 (1-)	1.65E+06(7=)	5.66E+05(5)
	S.D.	2.80E+05	1.98E+05	1.12E+05	7.29E+05	3.54E+04	5.04E+06	3.11E+05
F_{19}	*Mean*	2.92E+02 (1-)	1.17E+05(6=)	1.31E+04(4=)	1.34E+04(5=)	1.14E+04(2=)	1.43E+06(7+)	1.23E+04(3)
	S.D.	2.12E+02	6.70E+05	9.50E+03	9.67E+03	6.46E+03	4.59E+06	6.19E+03
F_{20}	*Mean*	8.04E+02(7+)	4.57E+02(2+)	6.70E+02(4+)	5.21E+02(3+)	7.38E+02(6+)	7.04E+02(5+)	2.86E+02 (1)
	S.D.	1.70E+02	2.01E+02	3.10E+02	2.27E+02	2.83E+02	2.68E+02	9.25E+01
F_{21}	*Mean*	4.82E+02(7+)	3.21E+02(4+)	2.81E+02(2+)	2.88E+02(3+)	3.25E+02(5+)	3.84E+02(6+)	2.38E+02 (1)
	S.D.	8.01E+01	2.50E+01	2.06E+01	2.14E+01	2.61E+01	3.66E+01	1.52E+01
F_{22}	*Mean*	5.74E+03(6+)	4.63E+03(4+)	3.83E+03(2+)	4.11E+03(3+)	5.13E+03(5+)	6.13E+03(7+)	4.05E+02 (1)
	S.D.	1.47E+03	1.91E+03	2.56E+03	2.71E+03	1.74E+03	1.01E+03	9.60E+02
F_{23}	*Mean*	7.48E+02(6+)	7.08E+02(5+)	5.11E+02(2+)	5.21E+02(3+)	6.03E+02(4+)	9.01E+02(7+)	4.52E+02 (1)
	S.D.	3.60E+01	1.09E+02	2.16E+01	3.16E+01	3.93E+01	1.32E+02	1.21E+01
F_{24}	*Mean*	1.03E+03(7+)	9.03E+02(5+)	5.77E+02(2+)	6.24E+02(3+)	6.93E+02(4+)	1.01E+03(6+)	5.25E+02 (1)
	S.D.	5.89E+01	1.48E+02	1.77E+01	6.93E+01	4.92E+01	1.43E+02	1.46E+01
F_{25}	*Mean*	5.56E+02(5+)	5.37E+02(2+)	5.51E+02(4+)	5.89E+02(6+)	5.43E+02(3+)	7.94E+02(7+)	4.83E+02 (1)
	S.D.	1.54E+01	3.99E+01	3.55E+01	3.44E+01	4.19E+01	3.21E+02	8.02E+00
F_{26}	*Mean*	2.23E+03(5+)	2.11E+03(4+)	1.72E+03(3+)	1.46E+03(2+)	2.64E+03(6+)	4.20E+03(7+)	3.85E+02 (1)
	S.D.	1.29E+03	9.32E+02	9.50E+02	9.29E+02	9.97E+02	1.05E+03	2.61E+02
F_{27}	*Mean*	6.49E+02(4+)	7.94E+02(6+)	6.04E+02(2+)	7.07E+02(5+)	6.36E+02(3+)	9.98E+02(7+)	5.24E+02 (1)
	S.D.	3.02E+01	1.68E+02	4.45E+01	8.29E+01	6.70E+01	2.13E+02	2.10E+01
F_{28}	*Mean*	5.36E+02(4+)	5.95E+02(6+)	4.98E+02(3+)	5.38E+02(5+)	4.88E+02(2+)	1.52E+03(7+)	4.68E+02 (1)
	S.D.	1.47E+01	1.59E+02	3.36E+01	3.14E+01	1.94E+01	9.49E+02	1.80E+01
F_{29}	*Mean*	1.08E+03(6+)	1.07E+03(5+)	7.63E+02(2+)	8.62E+02(3+)	9.57E+02(4+)	1.25E+03(7+)	5.68E+02 (1)
	S.D.	1.49E+02	2.59E+02	2.49E+02	2.48E+02	2.52E+02	3.73E+02	9.38E+01
F_{30}	*Mean*	9.68E+05(4+)	1.41E+06(5+)	8.16E+05(3+)	1.89E+06(6+)	7.42E+05 (1=)	4.94E+07(7+)	7.48E+05(2)
	S.D.	1.43E+05	9.41E+05	8.75E+04	5.18E+05	7.72E+04	2.22E+08	8.26E+04
(#)	+	26	26	21	21	20	27	-
(#)	-	3	2	5	2	4	0	-
Avg	(*Rank*)	5.37	4.40	2.63	3.77	3.40	6.67	1.77

tions in 30D, 50D, and 100D cases, respectively. Moreover, GLPSO and TAPSO also attain promising performance on the composition functions with higher dimension.

From the comparison results on 4 distinct types of functions we can observe that MFCPSO is more suitable for complicated and higher dimension problems. Moreover, the values of $Avg(Rank)$ verify that MFCPSO also attains the best performance on the CEC2017 test suite in 4 different dimension cases.

C. Statistic Results of Solutions

1) t-test Results: In this section, statistic results of the t-test between MFCPSO and other 6 peer algorithms in the

This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

TABLE VII: Comparison results of solution accuracy on CEC2017 test suite(D=100).

		CCPSO-ISM	SRPSO	GLPSO	XPSO	TAPSO	AWPSO	MFCPSO
F_1	*Mean*	7.10E+03(5+)	1.57E+09(6+)	3.40E+03 (1=)	6.35E+03(3+)	6.46E+03(4+)	3.34E+10(7+)	3.79E+03(2)
	S.D.	4.80E+03	1.96E+09	3.62E+03	7.21E+03	7.85E+03	1.59E+10	4.29E+03
F_2	*Mean*	2.58E+84(5=)	1.83E+106(6=)	1.95E+43 (1=)	2.43E+81(4=)	3.73E+56(2=)	3.10E+132(7=)	1.02E+58(3)
	S.D.	1.83E+85	1.31E+107	1.39E+44	1.73E+82	2.35E+57	2.21E+133	7.28E+58
F_3	*Mean*	4.88E+05(7+)	3.02E+04 (1-)	5.33E+04(3-)	9.35E+04(4-)	3.34E+04(2-)	9.91E+04(5-)	1.73E+05(6)
	S.D.	3.14E+04	7.12E+03	1.35E+04	1.08E+04	1.36E+04	2.70E+04	1.17E+04
F_4	*Mean*	3.01E+02(4+)	4.54E+02(5+)	2.12E+02(2-)	4.86E+02(6+)	1.74E+02 (1-)	3.42E+03(7+)	2.32E+02(3)
	S.D.	3.52E+01	3.65E+02	3.48E+01	6.84E+01	4.45E+01	1.74E+03	5.99E+00
F_5	*Mean*	8.05E+02(7+)	3.22E+02(4+)	2.14E+02(2+)	2.17E+02(3+)	3.58E+02(5+)	4.46E+02(6+)	1.20E+02 (1)
	S.D.	5.83E+01	7.16E+01	3.31E+01	3.76E+01	4.26E+01	6.55E+01	3.25E+01
F_6	*Mean*	4.78E+00(5+)	5.11E+00(6+)	1.58E-03(2+)	3.49E+00(4+)	7.17E-02(3+)	1.68E+01(7+)	3.82E-06 (1)
	S.D.	3.93E+00	2.97E+00	1.16E-03	3.20E+00	2.79E-02	3.92E+00	2.45E-06
F_7	*Mean*	6.38E+02(6+)	3.57E+02(3+)	3.22E+02(4+)	4.31E+02(2+)	4.96E+02(5+)	7.56E+02(7+)	2.79E+02 (1)
	S.D.	5.19E+01	5.16E+01	3.19E+01	9.68E+01	6.78E+01	2.09E+02	4.27E+01
F_8	*Mean*	7.65E+02(7+)	3.24E+02(4+)	2.05E+02(2+)	2.23E+02(3+)	3.39E+02(5+)	4.69E+02(6+)	1.22E+02 (1)
	S.D.	6.27E+01	5.59E+01	3.57E+01	5.00E+01	4.72E+01	8.20E+01	3.96E+01
F_9	*Mean*	6.20E+04(7+)	2.55E+03(4+)	9.93E+01(2+)	3.72E+02(3+)	5.85E+03(5+)	1.37E+04(6+)	1.76E-03 (1)
	S.D.	7.59E+03	3.97E+03	1.63E+02	2.93E+02	2.58E+03	1.12E+04	1.25E-02
F_{10}	*Mean*	1.42E+04(7+)	1.11E+04(2+)	1.11E+04(3+)	1.24E+04(5+)	1.21E+04(4+)	1.38E+04(6+)	6.40E+03 (1)
	S.D.	1.01E+03	1.48E+03	1.55E+03	1.14E+03	1.18E+03	1.29E+03	9.76E+02
F_{11}	*Mean*	5.51E+03(7+)	1.17E+03(5+)	2.46E+02 (1-)	1.16E+03(4+)	3.19E+02(2-)	4.15E+03(6+)	6.37E+02(3)
	S.D.	1.09E+03	5.97E+02	5.58E+01	1.83E+02	9.86E+01	3.28E+03	8.16E+01
F_{12}	*Mean*	2.04E+07(4+)	5.35E+08(6+)	1.20E+06(2-)	2.14E+07(5+)	5.13E+05 (1-)	8.38E+09(7+)	5.24E+06(3)
	S.D.	5.05E+06	1.19E+09	4.80E+05	2.26E+07	2.91E+05	6.67E+09	1.63E+06
F_{13}	*Mean*	5.23E+03(4=)	3.41E+07(6+)	3.86E+03 (1=)	4.11E+03(3=)	5.43E+03(5=)	1.13E+09(7+)	3.89E+03(2)
	S.D.	3.78E+03	1.14E+08	4.48E+03	3.85E+03	6.01E+03	1.64E+09	3.91E+03
F_{14}	*Mean*	2.21E+06(7+)	2.90E+05(3-)	6.99E+04(2-)	4.05E+05(4=)	3.30E+04 (1-)	1.51E+06(6+)	4.75E+05(5)
	S.D.	5.83E+05	1.42E+05	2.84E+04	5.37E+05	2.10E+04	1.34E+06	1.56E+05
F_{15}	*Mean*	1.88E+03(2=)	6.34E+03(6+)	2.78E+03(5+)	2.28E+03(3+)	2.64E+03(4+)	2.06E+08(7+)	1.30E+03 (1)
	S.D.	1.19E+03	7.26E+03	3.22E+03	2.56E+03	3.41E+03	3.73E+08	1.94E+03
F_{16}	*Mean*	3.14E+03(5+)	2.37E+03(2+)	2.94E+03(4+)	2.87E+03(3+)	3.22E+03(6+)	4.35E+03(7+)	1.35E+03 (1)
	S.D.	3.44E+02	5.58E+02	6.83E+02	5.94E+02	7.77E+02	7.62E+02	3.52E+02
F_{17}	*Mean*	2.43E+03(4+)	2.23E+03(3+)	2.11E+03(2+)	2.61E+03(6+)	2.51E+03(5+)	4.16E+03(7+)	9.28E+02 (1)
	S.D.	3.06E+02	4.57E+02	4.82E+02	5.28E+02	5.54E+02	6.31E+02	2.54E+02
F_{18}	*Mean*	2.45E+06(6+)	7.11E+05(3-)	1.03E+06(5+)	5.06E+05(2-)	1.09E+05 (1-)	3.35E+06(7+)	8.38E+05(4)
	S.D.	7.50E+05	3.73E+05	4.62E+05	2.38E+05	4.12E+04	2.58E+06	2.53E+05
F_{19}	*Mean*	1.78E+03 (1=)	8.36E+06(6+)	2.83E+03(3=)	3.00E+03(5+)	2.90E+03(4=)	1.80E+08(7+)	1.80E+03(2)
	S.D.	1.81E+03	2.50E+07	3.42E+03	3.67E+03	3.81E+03	2.76E+08	1.85E+03
F_{20}	*Mean*	2.51E+03(6+)	1.88E+03(2+)	2.19E+03(5+)	2.15E+03(4+)	2.14E+03(3+)	2.54E+03(7+)	1.13E+03 (1)
	S.D.	2.73E+02	4.23E+02	5.55E+02	6.08E+02	4.94E+02	6.68E+02	3.01E+02
F_{21}	*Mean*	1.00E+03(7+)	6.31E+02(5+)	4.32E+02(2+)	4.47E+02(3+)	5.95E+02(4+)	8.85E+02(6+)	3.43E+02 (1)
	S.D.	5.92E+01	6.14E+01	3.53E+01	4.08E+01	5.50E+01	1.18E+02	3.56E+01
F_{22}	*Mean*	1.59E+04(7+)	1.27E+04(3+)	1.15E+04(2+)	1.29E+04(4+)	1.33E+04(5+)	1.48E+04(6+)	6.67E+02 (1)
	S.D.	1.01E+03	2.63E+03	3.55E+03	4.85E+03	1.44E+03	1.79E+03	2.30E+03
F_{23}	*Mean*	1.02E+03(5+)	1.67E+03(6+)	6.78E+02(2+)	8.30E+02(4+)	8.10E+02(3+)	1.96E+03(7+)	5.91E+02 (1)
	S.D.	4.94E+01	2.82E+02	2.98E+01	5.76E+01	4.58E+01	2.27E+02	1.24E+01
F_{24}	*Mean*	1.53E+03(5+)	2.64E+03(6+)	1.07E+03(2+)	1.24E+03(3+)	1.35E+03(4+)	2.87E+03(7+)	9.10E+02 (1)
	S.D.	5.02E+01	5.16E+02	3.62E+01	9.48E+01	8.15E+01	4.10E+02	1.56E+01
F_{25}	*Mean*	8.15E+01(5+)	8.19E+02(4=)	7.62E+02 (1-)	1.08E+03(6+)	7.62E+02(2-)	1.78E+03(7+)	7.93E+02(3)
	S.D.	2.99E+01	8.58E+01	6.46E+01	8.51E+01	6.89E+01	6.06E+02	4.00E+01
F_{26}	*Mean*	8.96E+03(6+)	6.73E+03(4+)	5.16E+03(2+)	5.25E+03(3+)	7.40E+03(5+)	1.62E+04(7+)	1.18E+03 (1)
	S.D.	1.86E+03	2.30E+03	7.84E+02	1.54E+03	1.62E+03	2.66E+03	1.31E+03
F_{27}	*Mean*	7.64E+02(3+)	9.28E+02(6+)	7.33E+02(2+)	8.92E+02(5+)	7.82E+02(4+)	1.26E+03(7+)	5.76E+02 (1)
	S.D.	2.84E+01	2.07E+02	3.82E+01	6.73E+01	5.79E+01	2.93E+02	1.72E+01
F_{28}	*Mean*	6.73E+02(4+)	9.72E+02(6+)	5.75E+02(3+)	8.38E+02(5+)	5.44E+02 (1-)	5.28E+03(7+)	5.57E+02(2)
	S.D.	2.19E+01	6.99E+02	3.69E+01	5.23E+01	3.18E+01	2.62E+03	9.09E+00
F_{29}	*Mean*	3.33E+03(6+)	2.86E+03(3+)	2.53E+03(2+)	3.23E+03(5+)	3.14E+03(4+)	3.80E+03(7+)	1.64E+03 (1)
	S.D.	2.74E+02	4.14E+02	5.28E+02	6.06E+02	5.55E+02	5.33E+02	3.06E+02
F_{30}	*Mean*	3.75E+04(4+)	6.16E+07(6+)	4.95E+03 (1-)	3.85E+04(5+)	5.42E+03(2-)	8.78E+08(7+)	1.15E+04(3)
	S.D.	1.20E+04	1.81E+08	2.49E+03	3.58E+04	3.35E+03	8.99E+08	3.84E+03
(#)	+	26	25	19	25	18	28	-
(#)	-	0	3	7	2	9	1	-
Avg	*(Rank)*	5.27	4.40	2.30	4.03	3.40	6.67	1.93

4 dimension cases are displayed in Table VIII, in which symbols "(#)+" and "(#)-" denote the number that MFCPSO are significantly better than and significantly worse than the corresponding competitor algorithm, respectively. The comprehensive performance (*CP*) is equal to "(#)+" minus "(#)-" of all the 4 dimension cases.

It can be observed from Table VIII that MFCPSO significantly outperforms other 6 PSO variants on the all the test functions with 4 different dimension cases except that it is dominated by CCPSO-ISM and XPSO in 10D case. Moreover, according to the values of *CP*, we can see that MFCPSO exhibits the most favorable performance, followed by GLPSO,

TABLE VIII: t-test results between MFCPSO and other 6 PSO variants in 4 different dimension cases

Algorithm	10D		30D		50D		100D		*CP*
	(#)+	(#)-	(#)+	(#)-	(#)+	(#)-	(#)+	(#)-	
CCPSO-ISM	10	14	26	4	26	3	26	0	67
SRPSO	10	2	22	1	26	2	25	3	75
GLPSO	11	4	17	7	21	5	19	7	45
XPSO	5	7	19	2	21	2	25	2	57
TAPSO	16	4	17	5	20	4	18	9	49
AWPSO	17	2	29	0	27	0	28	1	98

TAPSO, and XPSO.

2) Friedman-test results: In this part, a set of Friedman-tests of *Mean* values is applied to compare the performance among all the 7 peer algorithms in 10D, 30D, 50D, and 100D cases. Furthermore, overall performance of the 4 dimension cases is also analyzed, in terms of the Friedman-test results. The results are listed in Table IX, in which each algorithm and its rankings are listed in ascending order (the lower the better).

The Friedman-test results verify that MFCPSO attains the best overall performance, followed by GLPSO, which is consistent with the result of the *t*-test listed in Table VIII. In addition, MFCPSO also achieves the most promising property in all the 4 different dimension cases while GLPSO yields the second outstanding performance in the 3 higher dimension cases, and XPSO attains the best result in the lowest dimension case. Furthermore, SRPSO, CCPSO-ISM, and AWPSO exhibit very unfavorable performance in the higher dimension cases. Although TAPSO does not offer promising performance in 10D and 30D cases, it exhibits much favorable characteristics in 50D and 100D cases, which verifies that the triple archives strategy applied in TAPSO may be very suitable for higher dimension problems.

D. Effectiveness of Parameters Adjusted by the Fuzzy Controller

In this part, effectiveness of the new introduced parameters is analyzed. From the experimental results we can observe that $c_{1,i}$ and p_i controlled by the MIMO fuzzy logic system endow MFCPSO with very promising characteristics. In this section, a set of experiments is conducted to analyze the effectiveness of the two parameters. Due to the space limitation, only 3 unimodal functions (F_1 - F_3) and 3 composition functions (F_{21} - F_{23}) are selected as test functions, and the dimension of them is *D*=30.

1) Effectiveness of $c_{1,i}$: As discussed in Section III-C that $c_{i,1}$ determines a learning weight that the particle *i* learns knowledge from an elite exemplar. In this part, performance of the fuzzy controlled $c_{i,1}$ is testified by comparison results between it and other 3 different constant values assigned to $c_{i,1}$.

From the experimental results demonstrated in Fig. 4 we can see that $c_{i,1}$ adjusted by the fuzzy controller is beneficial for the convergence speed before the middle search stage on the 3 unimodal functions. However, the population cannot be further improved during the later search stage, which causes the solution accuracy achieved by the fuzzy controlled $c_{i,1}$ is worsen than that by other 3 different values of $c_{i,1}$. On the contrary, the fuzzy controlled $c_{i,1}$ enables MFCPSO to attain the best solution accuracy on all the 3 composition functions, which have complicated properties. Moreover, MFCPSO also offers a faster convergence speed on F_{22} and F_{23}, while it displays a significant improvement at the end of search stage on F_{21}.

2) Effectiveness of p_i: In MFCPSO, the parameter p_i determines how many elites and mavericks should be selected as candidate exemplars. From the experimental results demonstrated in Fig. 5 we can observe that the population offers the highest convergence speed on the majority of the unimodal functions when p_i=0.2. The reason is that the smaller p_i causes only few elites are selected as candidates exemplars. In other words, only those elites with very better fitness can be chosen, which is favorable for speeding up the convergence, especially for unimodal functions. On the contrary, the fuzzy controlled p_i displays mediocre performance on both convergence speed and solution accuracy.

However, the population cannot yield a fast convergence speed on 2 out of the 3 composition functions during the initial search stage when p_i=0.2. The reason may be that complex fitness landscapes in the 3 complicated functions cause the population cannot finding the real optimal solution only relies on fewer elite exemplars. On the contrary, the fuzzy controlled p_i applied in MFCPSO offers the best solution accuracy. In addition, it also helps MFCPSO offer a fast convergence process.

V. Conclusion

In PSO community, it is a popular method that applying fitness as a criterion when selecting exemplars for a particle. Although the fitness-based selection mechanism plays a positive role in the exploitation ability, it sacrifices the exploration capability when optimizing some multimodal functions. In recent years, some novelty-based selection mechanisms enable a system immune to deceptions and local optima in some artificial life study due to that the selection mechanisms entirely ignore an objective of a specific problem.

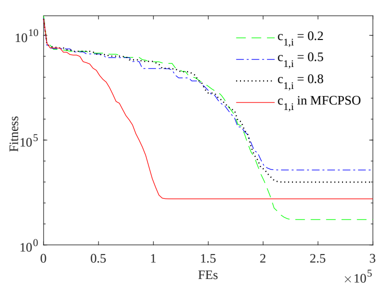
In this paper, to take advantages of the two selection mechanisms, a new PSO variant, name as MFCPSO, is proposed. In MFCPSO, the fitness-based selection and the novelty-based selection are used to select two types of candidate exemplars for a particle, which are separately named as elites and mavericks in this study. In each generation, the MIMO fuzzy logic controller is applied to tune two acceleration coefficients and a selection ratio for each particle. During the search process, two ideal control objectives (i.e., fitness and novelty) are dynamic adjusted. Based on the control objectives, each particle can adjust its own parameters by the fuzzy controller. As a result, not only each particle can perform distinct search behaviors during different search stages, but also different particles in a generation can display various search characteristics.

To testify the performance of MFCPSO, the CEC2017 test suite with 4 different dimension cases were selected as benchmark functions. From the comparison results between MFCPSO and other 6 PSO variants we can obtain some preliminary conclusions. Firstly, MFCPSO cannot offer very

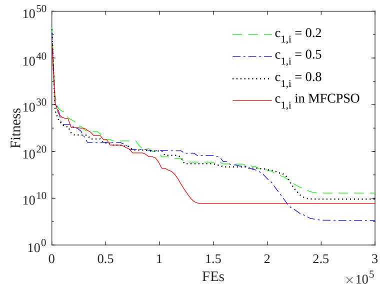
This article has been accepted for publication in IEEE Transactions on Fuzzy Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2022.3227464

TABLE IX: Friedman-test of mean values on the 30 test functions

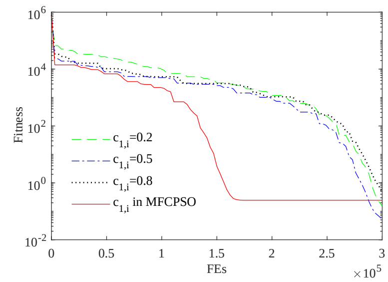
Overall Rank	Algorithm	Ranking	10D		30D		50D		100D	
			Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	MFCPSO	2.14	MFCPSO	3.05	MFCPSO	1.88	MFCPSO	1.72	MFCPSO	1.90
2	GLPSO	2.89	XPSO	3.13	GLPSO	2.58	GLPSO	2.69	GLPSO	2.34
3	XPSO	3.60	CCPSO_ISM	3.23	XPSO	3.52	TAPSO	3.47	TAPSO	3.43
4	TAPSO	3.95	GLPSO	3.93	TAPSO	4.00	XPSO	3.74	XPSO	4.03
5	SRPSO	4.36	SRPSO	4.18	SRPSO	4.53	SRPSO	4.34	SRPSO	4.36
6	CCPSO_ISM	4.66	TAPSO	4.87	CCPSO_ISM	4.78	CCPSO_ISM	5.38	CCPSO_ISM	5.28
7	AWPSO	6.40	AWPSO	5.60	AWPSO	6.70	AWPSO	6.66	AWPSO	6.66



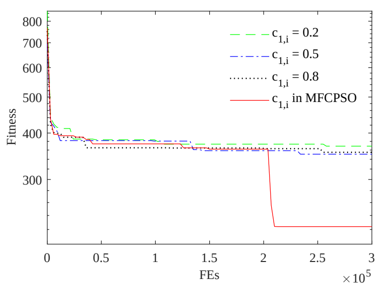
(a) F_1



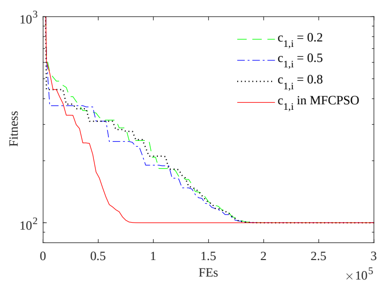
(b) F_2



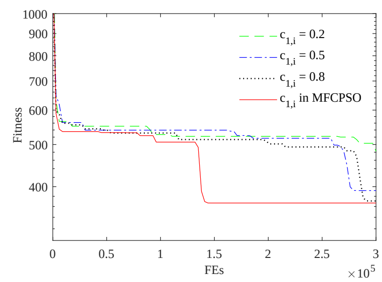
(c) F_3



(d) F_{21}

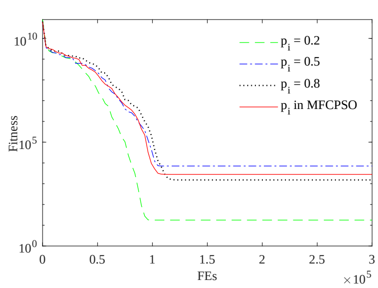


(e) F_{22}

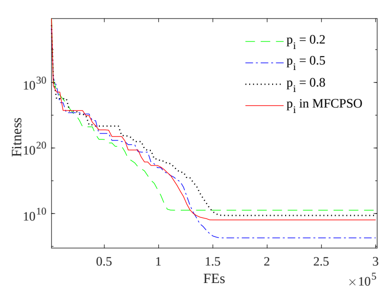


(f) F_{23}

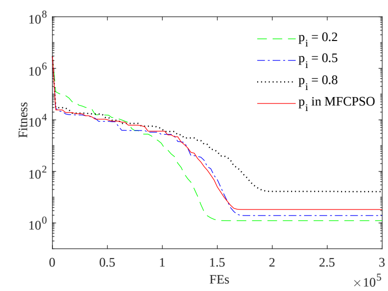
Fig. 4: Convergence process under the different values of $c_{1,i}$.



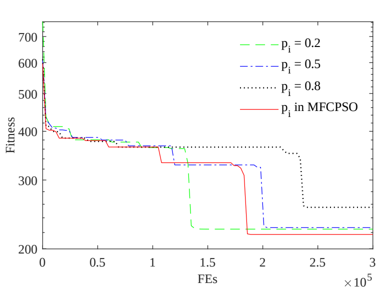
(a) f_1



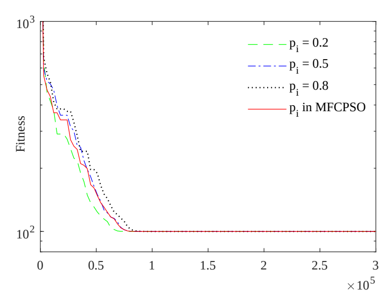
(b) f_2



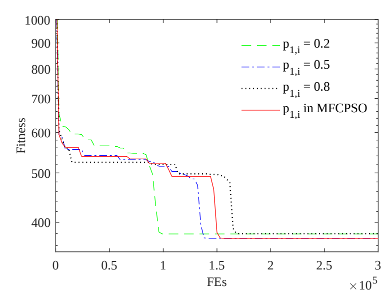
(c) f_3



(d) F_{21}



(e) F_{22}



(f) F_{23}

Fig. 5: Convergence process under the different values of p_i.

outstanding performance on simple unimodal functions in higher dimension cases though it achieves the most favorable performance on the unimodal functions in lower dimension case. Secondly, on the contrary, MFCPSO exhibits more promising performance on multimodal functions, especially on complicated multimodal functions in higher dimension cases. Lastly, MFCPSO attains the best overall performance on the CEC2017 test suite in different dimension cases.

Furthermore, the effectiveness of the proposed strategy also has been verified be a set of experiments. From the comparison results on $c_{1,i}$ and p_i we can obtain a few preliminary conclusions. Firstly, the fuzzy controlled $c_{1,i}$ is favorable for speeding up convergence on unimodal functions, especially during the middle search stage. Secondly, the $c_{1,i}$ can help MFCPSO obtain more accurate solutions on complicated multimodal functions. Lastly, the fuzzy controlled p_i is more suitable for the complicated functions than unimodal functions, which is similar as the fuzzy controlled $c_{1,i}$.

Although the new introduced novelty-based selection mechanism as well as the traditional fitness-based selection casts the performance of PSO in a new perspective, some issues need further study.

Firstly, how to define an effective criterion for "novelty" should be dealt with since it is crucial for a specific problem. In this paper, a particle's novelty is measured by an average distance between the particle and its the K-nearest neighbors. However, it is not to say that the definition is the optimal choice due to that different problems possess their own distinct properties. Hence, it is more realistic that designing an appropriate measurement for "novelty" relying on a problem's characteristics. In addition, some crucial factors in the MIMO fuzzy logic controller need to be further studied, including definitions of fuzzy rules, membership functions, and defuzzification strategy. Lastly, performance of MFCPSO on complicated real applications also needs to be verified. In fact, the optimization process of MFCPSO can be regarded as a response system for a dynamic environment. Based on the MIMO fuzzy logic controller, the optimization system can exhibit proper responses for the dynamic environment. Thus, we will further improved our study for some dynamic optimization problems, including dynamic logistics management and multi-UAVs cooperative coverage search, and so on. For instance, to enhance capabilities of MFCPSO on the complicated real applications, we regard that fuzzy rules and membership functions of a fuzzy logic controller should be dynamic adjusted, aiming to accurately reflect characteristics of dynamic environments. As a result, the optimization algorithm could satisfy distinct requirements of different optimization phases, and then exhibit very adaptive and reliable performance.

References

[1] B. Wei, W.S. Zhang, and X.W. Xia, "Efficient feature selection algorithm based on particle swarm optimization with learning memory," *IEEE Access,* vol. 7, pp. 166066-166078, 2019.

[2] P. Shafipour, A. Rashno, and S. Fadaei, "Particle distance rank feature selection by particle swarm optimization," *Expert Syst. Appl.*, vol. 185, Article. 115620, Dec. 2021.

[3] M. Gong, Q. Cai, and X.W. Cheng, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 82-97, Feb. 2014.

[4] W.B. Liu, Z.D. Wang, and Y. Yuan, "A novel sigmoid-function-based adaptive weighted particle swarm optimizer," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 1085-1093, Feb. 2021.

[5] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol.4, Perth, WA, Australia, 1995, pp.1942-1948.

[6] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. Int. Symp. Micromach. Human Sci.*, Nagoya, Japan, 1995, pp.39-43.

[7] X. Zhang, K.J. Du, and Z.H. Zhan, "Cooperative co-evolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.,* vol. 50, no. 10, pp. 4454-4468, Aug. 2019.

[8] Z.J. Wang, Z.H. Zhan, and W.J. Yu, "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE Trans. Cybern.,* vol. 50, no. 6, pp. 2715-2729, June 2020.

[9] M. Črepinšek, S.H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1-33, Jun. 2013.

[10] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, Anchorage, AK, May 1998, pp. 68-73.

[11] Y. Shi and R.C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Soul, Korea, 2001, pp. 101-106.

[12] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Comput. Oper. Res.*, vol. 33, no. 3, pp. 859-871, Mar. 2004.

[13] J. Lu, H. Hu, and Y. Bai, "Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and adaboost algorithm," *Neurocomputing*, vol. 152, pp. 305-315, Mar. 2015.

[14] M.R. Tanweer, S. Suresh, and N. Sundararajan, "Self regulating particle swarm optimization algorithm," *Inf. Sci.*, vol. 294, no. 10, pp. 182-202, Feb. 2015.

[15] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, May 2002, pp.1671-1676.

[16] Q.F. Liu, W.H. Wei, and H.Q. Yuan, "Topology selection for particle swarm optimization," *Inf. Sci.*, vol. 363, pp. 154-173, Oct. 2016.

[17] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler,maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204-210, Jun. 2004.

[18] N. Kumar, A.A. Shaikh, and S.K. Mahato, "Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations," *Expert Syst. Appl.*, vol. 172, Article ID: 114646. 2021.

[19] Y. Bi, A. Lam, and H. Quan, "A comprehensively improved particle swarm optimization algorithm to guarantee particle activity," *Russ. Phys. J.*, vol. 64, pp. 866-875, 2021.

[20] R. Standish, "Open-ended artificial evolution," *Int. J. Comput. Intell. Appl.*, vol. 3, no. 2, pp. 167-175, 2003.

[21] J. Gomes, P. Urbano, and A.L. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intell.*, vol. 7, no. 2-3, pp. 115-144, 2013.

[22] J. Lehman and K.O. Stanley, "Efficiently evolving programs through the search for novelty," in *Proc. Genet. Evol. Comput. Conf.,* GECCO'10. Portland, Oreg., 2010, pp. 837-844.

[23] J. Lehman and K.O. Stanley, "Abandoning objectives: evolution through the search for novelty alone," *Evol. Comput.*, vol. 19, no. 2, pp. 189-223, June 2011.

[24] A. Ratnaweera, 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, June 2004.

[25] C. Pornsing, M.S. Sodhi, and B.F. Lamond, "Novel self-adaptive particle swarm optimization methods," *Soft Comput.*, vol. 20, no. 9, pp. 3579-3593, Sep. 2016.

[26] A.A. Nickabadi, M.M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3658-3670, Jun. 2011.

[27] Z.H. Zhan, J. Zhang, and Y. Li, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009.

[28] M.R. Tanweer, S. Suresh, and N. Sundararajan, "Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems," *Inf. Sci.*, vol. 326, pp. 1-24, Jan. 2016.

[29] X.W. Xia, Y. Xing, and B. Wei, "A fitness-based multi-role particle swarm optimization," *Swarm Evol. Comput.*, vol. 44, pp. 349-364, Feb. 2019.
[30] M.R. Tanweer, R. Auditya, and S. Suresh, "Directionally driven self-regulating particle swarm optimization algorithm," *Swarm Evol. Comput.*, vol. 28, pp. 98-116, June 2016.
[31] X.W. Xia, L. Gui, and G.L. He, "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," *Inf. Sci.*, vol. 508, pp. 105-120, 2020.
[32] K. Yasuda and N. Iwasaki, "Adaptive particle swarm optimization using velociy information of swarm," in *Proc. of IEEE Int. Conf. Syst. Man Cyber.*, Hague, Netherlands, 2004, pp. 3475-3481.
[33] J. Li, J.Q. Zhang, and C.J. Jiang, "Composite particle swarm optimizer with historical memory for function optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350-2363, Oct. 2015.
[34] J.J. Liang, A.K. Qin, and P.N. Suganthan, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol.10, no.3, pp. 281-295, Jun. 2006.
[35] Z.H. Zhan, Y. Li, and Y.H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832-847, Dec. 2011.
[36] Q.D. Qin, C. Shi, and Q. Zhang, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238-2251, Oct. 2016.
[37] G. Xu, Q. Cui, and X. Shi, "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.* vol. 45, pp. 33-51, Mar. 2019.
[38] 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.
[39] B. Inden, Y.C. Jin, and R. Haschke, "An examination of different fitness and novelty based selection methods for the evolution of neural networks," *Soft Comput.*, vol. 17, pp. 753-767, Nov. 2013.
[40] A.L. Guiffrida and R. Nagi, "Fuzzy set theory applications in production management research: A literature survey," *J. Intell. Manuf.*, vol. 9, no. 1, pp. 39-56, 1998.
[41] M.C.M. Teixeira and S.H. Zak, "Stabilizing controller design for uncertain nonlinear systems using fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 133-142, Apr. 1999.
[42] L. Liu, Z. Wang, and H. Zhang, "Data-based adaptive fault estimation and fault-tolerant control for MIMO model-free systems using generalized fuzzy hyperbolic model," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3191-3205, Dec. 2018.
[43] L. Liu, Y.J. Liu, and S. Tong, "Fuzzy-based multierror constraint control for switched nonlinear systems and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 8, pp. 1519-1531, Aug. 2019.
[44] L. Liu, Y.J. Liu, and A.Q., "Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems." *Sci. China Inf. Sci.*, vol. 63, no.3, Article. 132203, Mar. 2020.
[45] D. Chen, F. Zou, and R.Q. Lu, "A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization," *Swarm Evol. Comput.*, vol. 43, pp. 147-165, 2018.
[46] M. Orouskhani, D. Shi, and X. Cheng, "A fuzzy adaptive dynamic NSGA-II with fuzzy-based borda ranking method and its application to multimedia data analysis," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 1, pp. 118-128, Jan. 2021.
[47] S.C. Huang, M.K. Jiau, and C.H. Lin, "Optimization of the carpool service problem via a fuzzy-controlled genetic algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1698-1712, Oct. 2015.
[48] P. Dziwiński, and Ł. Bartczuk, "A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 6, pp. 1140-1154, June 2020.
[49] N.H. Awad, M.Z. Ali, and J.J. Liang, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technological Univ., Singapore, Tech. Rep., 2016.
[50] Y.H. Li, Z.H. Zhan, and S. Lin, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," *Inf. Sci.*, vol. 293, no. 3, pp. 370-382, Feb. 2015.
[51] Y.J. Gong, J.J. Li, and L. Yun, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277-2290, Oct. 2016.
[52] X.W. Xia, L. Gui, and F. Yu, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862-4875, Dec. 2020.

Xuewen Xia received the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2009.

In 2009, he was a lecturer with the Hubei Engineering University, Xiaogan, China. In 2012, he worked as a postdoctoral researcher at Wuhan University, Wuhan, China. In 2014, he was an associate professor with the School of Software, East China Jiaotong University, Nanchang, China. He is currently a professor with the College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China. His current research interests include the areas of computational intelligence techniques and their applications.

Haojie Song In 2020, Haojie Song received the baccalaureate degree in mechanics from Shandong Jianzhu University, Jinan,China. He is currently pursuing a master's degree at Minnan Normal University, Zhangzhou, China. His research interests include deep learning neural network and its applications.

Ling Gui In 2014, she was a laboratory technician with the Economics and Management, East China Jiaotong University, Nanchang, China. She is currently a laboratory technician with the College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China. Her current research interests include swarm intelligence techniques and their applications.

Xing Xu received the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2010. He is currently a professor with the College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China. His research interests include intelligent computation and blockchain technology.

Ying-Long Zhang received his PhD from RenMin University, China in 2014. He is currently an assistant professor with the College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China. His research interests include data mining and information network analysis.