

Limit-Cycle-Based Mutant Multiobjective Pigeon-Inspired Optimization

Haibin Duan^{id}, Senior Member, IEEE, Mengzhen Huo^{id}, and Yuhui Shi^{id}, Fellow, IEEE

Abstract—This article presents a limit-cycle-based mutant multiobjective pigeon-inspired optimization (PIO). In this algorithm, the limit-cycle-based mechanism is devised to consider the factors that affect the flight of pigeons to simplify the multiobjective PIO algorithm. The mutant mechanism is incorporated to strengthen the exploration capability in the evolutionary process. Additionally, the application of the dual repository makes the nondominated solutions stored and selected to guide the flight of pigeons. Attributed to the limit-cycle-based mutant mechanisms, this algorithm not only obtains the faster convergence speed and higher accuracy but also improves its population diversity. To confirm the universal application of this algorithm, theoretical analysis of the convergence is discussed in this article. Finally, comparative experiments of our proposed algorithm and other five multiobjective methods are conducted to verify the accuracy, efficiency, and convergence stability of the proposed algorithm.

Index Terms—Limit-cycle-based mechanism, multiobjective pigeon-inspired optimization (PIO), mutant mechanism, theoretical analysis.

I. INTRODUCTION

OPTIMIZATION problems are presented to solve real-life decisions and planning situations. However, there is always more than one objective to be optimized simultaneously in many scientific and engineering applications. These problems are termed as multiobjective optimization problems (MOPs).

Due to the conflicting property of multiple objectives, a single solution which can find the optimum for all the objectives

Manuscript received December 3, 2019; revised February 22, 2020; accepted March 16, 2020. Date of publication March 27, 2020; date of current version October 1, 2020. This work was supported in part by the Science and Technology Innovation 2030-Key Project of “New Generation Artificial Intelligence” under Grant 2018AAA0102303 and Grant 2018AAA0102403, in part by the National Natural Science Foundation of China under Grant 91948204, Grant 61761136008, Grant U1913602, Grant U19B2033, and Grant 91648205, and in part by the Aeronautical Foundation of China under Grant 20185851022. (Corresponding author: Haibin Duan.)

Haibin Duan is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China, and also with Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: hbduan@buaa.edu.cn).

Mengzhen Huo is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China, and also with the Shenyuan Honors College, Beihang University, Beijing 100083, China (e-mail: mzhuo@buaa.edu.cn).

Yuhui Shi is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: shiyh@sustech.edu.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2020.2983311

at the same time does not exist. The improvement of one objective function may result in the deterioration of others [1]. Thus, MOPs are aimed to find a set of tradeoff solutions (Pareto-optimal solutions) [2], known as Pareto set (PS) in the decision space and Pareto front (PF) in the objective space. The obtained Pareto-optimal solutions can be applied to various cases in real life.

Since the number of Pareto-optimal solutions is usually far more than the desired number, lots of multiobjective algorithms are represented to obtain a representative and diverse set of solutions for selection [3], [4]. Due to the generality and population-based searching nature, multiobjective evolutionary algorithms (MOEAs) could find the approximate set of Pareto-optimal solutions effectively and efficiently in a single run [5], [6]. Therefore, MOEAs have experienced great development [7] and played the major role in addressing complicate MOPs in the EA field. The most excellent representative MOEAs include the nondominated sorting genetic algorithm (NSGA) [8], MOEA based on decomposition (MOEA/D) [9], the strength Pareto evolutionary algorithm (SPEA) [10], and their variants, such as NSGA-II [11], NSGA-III [12], [13], MOEA/D-KF [14], SPEA-II [15], and other algorithms.

Many multiobjective evolutionary optimization algorithms generally have difficulties in solving large-scale multiobjective problems (LSMOPs). As the dimension of decision variables increases, the decision space usually exponentially grows and the general characteristics of the problem become more complicated which result in the premature convergence to local optima of MOEAs due to the degradation of population diversity in early stages of the optimization process [16]. Thus, nature-inspired heuristic algorithms, including evolutionary algorithms (EAs) [11], [17], artificial immune algorithms [18], [19], particle swarm optimization algorithms [20], [21], and pigeon-inspired optimization (PIO) algorithms [22] have been developed to tackle MOPs. Nevertheless, on the one hand, the performance of EAs always severely degrades when solving large-scale optimization problems, like DE algorithm [16]. For another, various strategies utilized in the improvement of MOEAs could bring the disadvantages of heavy calculation burden [23].

PIO is a newly proposed bio-inspired swarm intelligence algorithm, which was invented by Duan and Qiao [22] and successfully applied for solving real-world problems in various fields [24]–[26]. The basic algorithm of PIO is inspired by the behavior of the homing pigeons. In the evolutionary process of the algorithm, the pigeons could employ different

tools to guide the route during different periods. Many variants of PIO algorithm have been proposed to solve SOPs, and which are proven competitive in the quick speed of convergence and superior optimal solutions. The excellent performance of PIO in solving SOPs demonstrated its efficiency and effectiveness of searching the optimal solutions in complicate problems.

However, to improve the performance of the multiobjective PIO algorithm (MOPIO), it is necessary to design the corresponding mechanisms based on the topologies of its Pareto optimal solutions. There are two key problems to be solved. First, the relationship between the global exploration and local exploitation of the algorithm should be balanced, which is considered by almost all algorithms. Second, the MOPIO algorithm cannot follow the original calculation mechanism of the central pigeon in the PIO algorithm which needs to be considered.

Therefore, this article mainly focuses on these two respects and aims to extend PIO for tackling MOPs with high dimensions. We propose a limit-cycle-based mutant multiobjective PIO (CMMOPIO) algorithm. The major contributions of this article are listed as follows.

- 1) The limit cycle phenomenon has been applied in plenty of control fields with good control effect due to its dynamic stability. The limit-cycle-based mechanism is carried out to balance the global exploration ability and local exploitation ability by controlling the two parameters that affect the flight of pigeons in this article. Under the limit-cycle-based mechanism, the evolutionary trend of two parameters is consistent with the flight mechanism of the flock of pigeons.
- 2) To migrate the landmark operator from the PIO algorithm to the MOPIO algorithm, the mutant mechanism makes effort to select the central pigeon of the population from the nondominated solutions, so that it is not possible for the population to deviate from the true PF.
- 3) The superiority of the proposed algorithm compared with other existing MOPs is discussed through various experiments in this article. What is more, the convergence of the algorithm has been theoretically demonstrated, which proves the universality of this algorithm in solving MOPs.

The remainder of this article is organized as follows. Section II gives the related background of the proposed algorithm, including the mathematical description of MOPs and the introduction of the basic PIO algorithm. Section III gives the basic idea of the CMMOPIO algorithm, which includes limit-cycle-based mechanism, mutant mechanism, and dual repository. Section IV discusses the convergence of CMMOPIO and gives the convergence conditions. The results of comparative experiments are presented in Section V. Finally, our concluding remarks are given in Section VI.

II. BACKGROUND

A. Multiobjective Optimization Problems

Without loss of generality, MOPs involve multiple minimizing or maximizing conflicting objective functions. Supposing

m objective functions, an MOP can be stated as

$$\begin{aligned} \min / \max \quad & \mathbf{F}(\mathbf{x}) = [f_1(x), f_2(x), \dots, f_m(x)] \\ \text{Subject to} \quad & \mathbf{x} = [x_1, x_2, \dots, x_n] \in \Omega \end{aligned} \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]$ is the decision variable vector, and Ω is the decision variable space, $\mathbf{F} : \Omega \rightarrow R^m$, \mathbf{F} is the vector of m objective functions. Due to the conflict between the objectives, it is impossible to find one solution that can optimize the whole set of functions simultaneously. Thus, the definition of Pareto optimality is adopted to find out the best tradeoffs among all the functions.

Definition 1 (Pareto Dominance): A decision vector $\mathbf{y} = (y_1, y_2, \dots, y_k)$ is said to strictly dominate another decision vector $\mathbf{z} = (z_1, z_2, \dots, z_k)$ in minimization MOP, denoted by $\mathbf{y} < \mathbf{z}$. If and only if \mathbf{y} is partially less than \mathbf{z}

$$\forall i \in \{1, 2, \dots, k\}, \mathbf{y}_i \leq \mathbf{z}_i \wedge \exists i \in \{1, 2, \dots, k\} : \mathbf{y}_i < \mathbf{z}_i. \quad (2)$$

Definition 2 (Pareto Optimality): A point $\mathbf{x}^* \in \Omega$ is Pareto optimal if and only if for every $\mathbf{x} \in \Omega$

$$\neg \exists \mathbf{x} \in \Omega : \mathbf{x} \succ \mathbf{x}^*. \quad (3)$$

Definition 3 (Pareto Optimal Set): For a given MOP $\mathbf{f}(x)$, the Pareto optimal set \mathbf{P}^* is defined as

$$\mathbf{P}^* := \{x \in \Omega \mid \neg \exists x' \in \Omega, \mathbf{f}(x') \leq \mathbf{f}(x)\}. \quad (4)$$

Definition 4 (Pareto-Optimal Front): For a given MOP $\mathbf{f}(x)$ and Pareto optimal set \mathbf{P}^* , the PF (\mathbf{PF}^*) is defined as

$$\mathbf{PF}^* := \{\mathbf{y} = \mathbf{f} = (f_1(x), f_2(x), \dots, f_k(x)) \mid x \in \mathbf{P}^*\}. \quad (5)$$

In general, it is impossible to find the analytic expressions of the line or surface containing the points. Thus, the general procedure to generate the PF is to calculate the feasible points x and the corresponding $\mathbf{f}(x)$. When sufficient quantities are available, it is possible to determine the nondominated points and generate the PF.

B. Basic Pigeon-Inspired Optimization

PIO is a population-based bio-inspired swarm intelligence optimization algorithm, in which the individual is referred as a pigeon. In this algorithm, two operators (map and compass operator, landmark operator) are employed to guide the pigeons to find the best positions, the solutions of the problem to be solved.

1) *Map and Compass Operator:* For the optimization problem, each pigeon has a position, which is defined as

$$\mathbf{X}_i = [x_{i1}, x_{i2}, \dots, x_{iD}] \quad (6)$$

where D is the dimension of the function to be solved, $i = 1, 2, \dots, N_p$ is the population of the pigeons. Each pigeon has the velocity, which is stated as

$$\mathbf{V}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]. \quad (7)$$

First, randomly initialize the pigeon's position and velocity within the searching space. Then, with the number of the iterations increasing, the \mathbf{V}_i of the next iteration can be updated by

$$\mathbf{V}_i(t) = \mathbf{V}_i(t-1) \cdot e^{-Rt} + \text{rand} \cdot (\mathbf{X}_{\text{gbest}} - \mathbf{X}_i(t-1)) \quad (8)$$

where t is the current number of the iteration, $t = 1, 2, \dots, T_{1\max}$, $T_{1\max}$ is the maximum number of the iterations in which the map and compass operator works. Map and compass operator R is the weight factor used to control the speed of the iteration on the next iteration, which is the random number value uniformly distributed between 0 and 1. $\mathbf{X}_{\text{gbest}}$ is the global optimal solution during the current iterations. Then the next \mathbf{X}_i is calculated as follows:

$$\mathbf{X}_i(t) = \mathbf{X}_i(t-1) + \mathbf{V}_i(t). \quad (9)$$

The first evolutionary procedure utilizing the map and compass operator will be stopped until the iteration equaling to $T_{1\max}$.

2) *Landmark Operator*: As the pigeons get close to the destination, they gradually reach home depending on the nearby landmarks. In the landmark operator, the pigeons that are not familiar with the landmarks, will fly away from the correct direction and be discarded. Thus, the number of the pigeons is decreased by half after each iteration. The rest pigeons will utilize the landmark operator as navigation guidance, which is defined as $\mathbf{X}_{\text{center}}$. Hence, the position at the next iteration can be updated by

$$\mathbf{X}_i(t) = \mathbf{X}_i(t-1) + \text{rand} \cdot (\mathbf{X}_{\text{center}}(t-1) - \mathbf{X}_i(t-1)) \quad (10)$$

$$\mathbf{X}_{\text{center}}(t) = \frac{\sum_{i=1}^N \mathbf{X}_i(t) \cdot \text{fitness}(\mathbf{X}_i(t))}{N_p \cdot \sum_{i=1}^N \text{fitness}(\mathbf{X}_i(t))} \quad (11)$$

$$N_p(t) = \text{ceil}\left(\frac{N_p(t-1)}{2}\right) \quad (12)$$

where t is the current number of the iteration, $t = 1, 2, \dots, T_{2\max}$, $T_{2\max}$ is the maximum number of the iterations in which the landmark operator works. The fitness is the function to be optimized

$$\text{fitness}(\mathbf{X}_i(t)) = f_{\max}(\mathbf{X}_i(t)), \text{ for maximum problems} \quad (13)$$

$$\text{fitness}(\mathbf{X}_i(t)) = \frac{1}{f_{\min}(\mathbf{X}_i(t)) + \varepsilon}, \text{ for minimum problems.} \quad (14)$$

After each iteration, the pigeon's position will be close to the center position until the iterations reach $T_{2\max}$.

III. PROPOSED ALGORITHM

A. Main Algorithm

The procedure of CMMOPIO algorithm is presented in Algorithm 1.

B. Limit-Cycle-Based Mechanism

The basic PIO algorithm employs two independent useful cycles to mimic the characteristic of the homing pigeons. The map and compass operator will merge during the later period as the landmark operator works for the navigation of homing pigeons. In our proposed algorithm, these two operators are combined in one process under the limit-cycle-based mechanism. Considered a limit cycle, it has the property that all variables in the vicinity of the limit cycle ultimately tend

Algorithm 1 CMMOPIO Algorithm Procedure

-
- 1: **For** $i = 1$ **to** N_p **do**
 - 2: Initialize the position \mathbf{X} and velocity \mathbf{V} of each pigeon
 - 3: Evaluate the function values \mathbf{F} of each pigeon
 - 4: **End for**
 - 5: Get non-dominated pigeon individuals for $i = 1, \dots, N_p$
 - 6: Create D-dimension hypercubes and put the pigeons into the corresponding positions
 - 7: Initialize the best position of each pigeon $\mathbf{X}_{\text{gbest}} = \mathbf{X}(1)$
 - 8: **For** $t = 1$ **to** T_{\max}
 - 9: Calculate the weight coefficients k_1 and k_2 with the limit-cycle-based mechanism
 - 10: Calculate the $\mathbf{X}_{\text{center}}^t$ with the mutant mechanism.
 - 11: Update the velocity $\mathbf{V}(t)$ and position $\mathbf{X}(t)$:

$$\mathbf{V}_i^{t+1} = e^{-Rt} \cdot \mathbf{V}_i^t + k_1 \cdot (\mathbf{X}_{\text{gbest},i}^t - \mathbf{X}_i^t) + k_2 \cdot (\mathbf{X}_{\text{center}}^t - \mathbf{X}_i^t) \quad (15)$$

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{V}_i^{t+1} \quad (16)$$
 - 12: Border check: If the variables go beyond their upper boundary, then make it the maximum boundary value. Vice versa.
 - 13: Evaluate each of the pigeons in population and update the best position of each pigeon
 - 14: Update the secondary repository
 - 15: **End for**
-

toward the limit cycle as time goes into infinity. Following this idea, we utilize the limit-cycle-based mechanism to update two operators in (15). k_1 and k_2 are defined by follows:

$$\begin{bmatrix} k_{1,i}^{t+1} \\ k_{2,i}^{t+1} \end{bmatrix} = \begin{bmatrix} k_{1,i}^t \\ k_{2,i}^t \end{bmatrix} + L_i^t \quad (17)$$

$$L_i^t = \lambda \cdot \begin{bmatrix} \gamma_i^t & -1 \\ 1 & \gamma_i^t \end{bmatrix} \cdot \begin{bmatrix} k_{1,i}^t \\ k_{2,i}^t \end{bmatrix} \quad (18)$$

$$\gamma_i^t = 1 - (k_{1,i}^t)^2 - (k_{2,i}^t)^2. \quad (19)$$

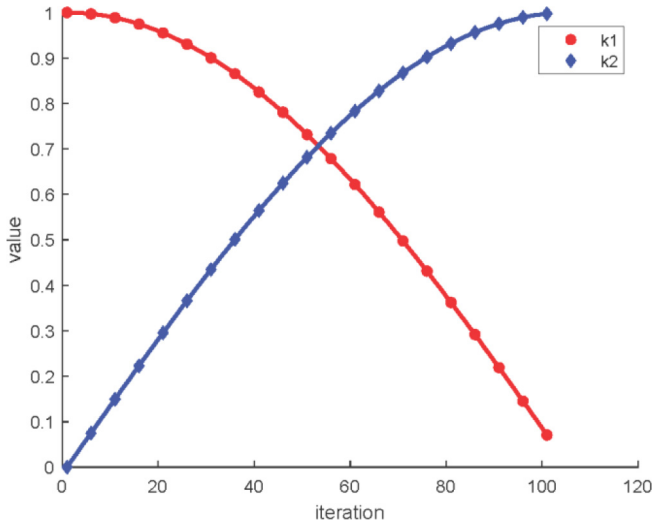
As two operators of PIO algorithm work in different periods during the flight of pigeons, the tendency of their weight coefficients k_1 and k_2 needs to be corresponding to the pigeons homing behavior. At the first period, the $\mathbf{X}_{\text{gbest}}$ of pigeons plays a major role in the flock, thus k_1 is usually initialized as 1. At the second period, the effect of the $\mathbf{X}_{\text{gbest}}$ descends and the effect of the $\mathbf{X}_{\text{center}}^t$ increases beyond that of the $\mathbf{X}_{\text{center}}^t$. Thus, the coefficient of the $\mathbf{X}_{\text{center}}^t$, k_2 , is initially set to be 0 and gradually increases. The evolution of two parameters under the limit-cycle-based mechanism is shown in the following Fig. 1.

As we can see from the two equations above, k_1 is bigger than the k_2 at the beginning. As the number of t increases, k_1 gradually decreases and k_2 increases until k_2 is over k_1 . Thus, the best position of the pigeon $\mathbf{X}_{\text{gbest},i}^t$ plays the leading role at first and the $\mathbf{X}_{\text{center}}^t$ guides the navigation later.

C. Mutant Mechanism

As referred in the part A, $\mathbf{X}_{\text{center}}^t$ is calculated based on the mutant mechanism. In basic PIO algorithm, $\mathbf{X}_{\text{center}}^t$ is obtained by the (11) utilizing the maximum or the minimum value of the solutions. While there is no one solution superior to all the others in solving multiobjective problems, it is important to design a mutant mechanism to generate the $\mathbf{X}_{\text{center}}^t$.

Due to the successful application of mutation operator in MOEAs, our proposed algorithm employs the mutant mechanism to improve the exploitation and applicability capability

Fig. 1. Evolutionary curve of parameters $k1$ and $k2$.**Algorithm 2** Mutant Mechanism Procedure

```

1: Initialize  $m = 20$ , the upper and lower bound  $ub, lb$ 
2: select the non-dominated solution randomly  $\mathbf{X}_{\text{REP}}(j)$  in secondary
   repository
3: if  $\text{rand} < 1/m$ 
4:    $a(k) = 1$ 
5: else
6:    $a(k) = 0$ 
7: end if
8: calculate  $\Delta d$ , calculate  $\mathbf{X}_{\text{center},i}^t$ 
9: calculate the new function value  $F_{\text{New}}$ 
10: if  $F_{\text{New}}$  dominates  $F_{\text{Old}}$ 
11:   update  $\mathbf{X}_{\text{center},i}^t$ 
12: end if

```

of solving multiobjective problems as the number of the iteration increasing. $\mathbf{X}_{\text{center},i}^t$ is generated based on the non-dominated solutions. Then, the mutant mechanism is employed to improve the chosen solution based on the step mutation operator in (20) and (21). The chosen solutions are expanded in D dimensions, respectively. While the solution with certain dimension mutated could obtained the better function value, then the $\mathbf{X}_{\text{center},i}^t$ is updated. The mutant mechanism procedure is introduced in Algorithm 2

$$\mathbf{X}_{\text{center}}^t(i, j) = X_{\text{Rep}}(j) + \Delta d \cdot (ub - lb) \quad (20)$$

$$\Delta d = \text{sum} \left(\frac{a(k)}{2^k} \right) a(k) = \begin{cases} 0, & \text{rand} < 1/m \\ 1, & \text{rand} \geq 1/m \end{cases} \quad (21)$$

where $k = 0, 1, 2, \dots, m$.

D. Dual Repository

There are two repositories in our proposed CMMOPIO algorithm. One of the repositories is consisting of the pigeons' positions of the current iteration, and the other one of the repositories is used to keep the record of the nondominated positions chosen from the evolutionary process. The secondary repository is designed in [27], which is composed of the archive controller and the grid.

These two components are introduced in the following.

1) *The Archive Controller*: The effect of the archive controller is to decide whether the new updated solution (S)

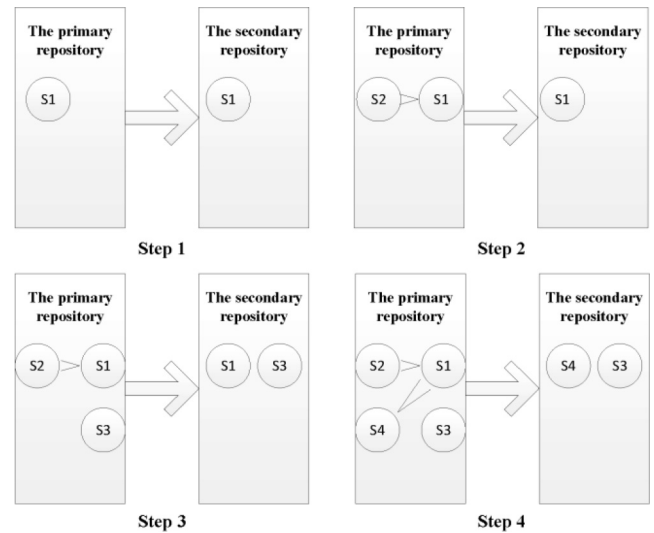


Fig. 2. Steps of the archive controller.

should be saved in the archive or not. The process of making the choice is as follows.

The nondominated solutions obtained in the evolutionary process are compared with the individuals in the secondary repository, which is empty at the beginning of the search. If the secondary repository is empty, then the new nondominated decision $S1$ is absorbed in (see step 1, in Fig. 2). If the new decision $S2$ is dominated by the individual $S1$ within the secondary repository, then the new solution $S2$ is removed automatically (see step 2, in Fig. 2). While there are no individuals in the secondary repository that dominate the new solution $S3$, then the new solution is stored in the secondary repository (see step 3, in Fig. 2). If the new solution $S4$ dominates the solution $S1$ in the secondary repository, as well as not dominated by other solutions in the secondary repository, then the solution $S4$ is absorbed in the secondary repository and the solution $S1$ is kicked out at the same time (see step 3, in Fig. 2). When the number of the individuals in the secondary repository reaches the maximum allowable capacity, then the grid procedure is started.

2) *The Grid Procedure*: The effect of the grid procedure is to distribute the large number of solutions in the secondary repository in a uniform way.

To make the PFs distributed uniformly, a variation of the adaptive grid proposed in [28] are applied. The basic principle is to utilize the secondary repository to maintain the solutions that are nondominated with respect to the individuals in the primary repository. In the secondary repository, the search space of the multiobjective problem is divided into square grids as Fig. 3.

Principle 1: When the secondary repository has a full capacity, if the new nondominated solution $S0$ is stored in, then we move the solution located in the dense regions out like $S9$ as Fig. 4.

Principle 2: When the secondary repository has a full capacity, if the new nondominated solution $S0$ is located out of the current space, then we recalculate the bounds of the grid and remove the certain solutions as principle 1 as Fig. 5.

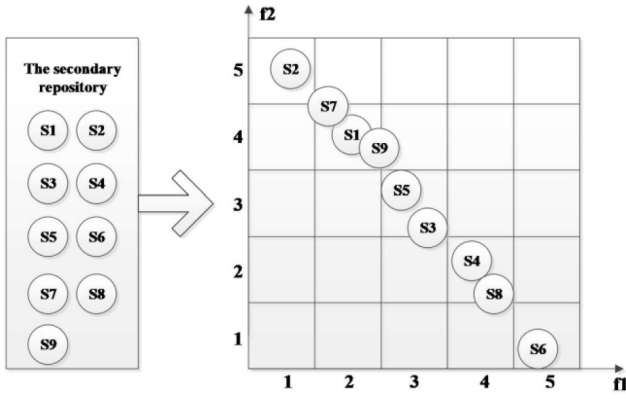


Fig. 3. Distribution of the secondary repository.

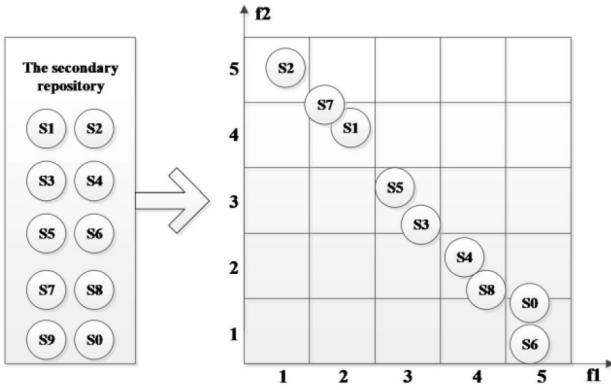


Fig. 4. Principle 1.

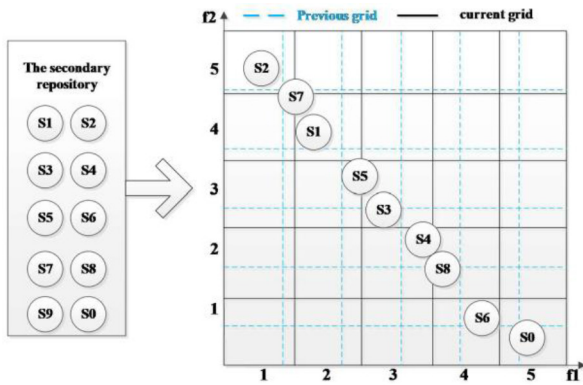


Fig. 5. Principle 2.

IV. CONVERGENCE ANALYSIS

The convergence properties of the PIO algorithm have been widely studied by researchers. However, the discussion of the convergence of the MOPIO is quite scarce. In this section, the convergence of the CMMOPIO algorithm is analyzed in the search space to get the convergence conditions. This analysis is based on the concept of Pareto optimality. To discuss the convergence of the CMMOPIO algorithm, some assumptions are needed as stated in the following.

Assumption 1: $\mathbf{X}_{\text{gbest},i}$ and $\mathbf{X}_{\text{center}}$, defined in Section II, satisfy the condition $\{\mathbf{X}_{\text{gbest},i}, \mathbf{X}_{\text{center}}\} \in \Gamma$, where Γ is the search space, $i = 1, 2, \dots, N_P$.

Assumption 2: There exists Pareto optimal set \mathbf{X}^* for $\mathbf{X}_i(t)$, $i = 1, 2, \dots, N_P$.

Assumption 3: The parameters k_1 and k_2 , defined in Section III-B, satisfy the condition that $0 < k_1 + k_2 < 2 \cdot (1 + e^{-Rt})$.

Theorem 1: When Assumption 1–3 are satisfied, given $k_1 \geq 0$, $k_2 \geq 0$, the pigeons' position \mathbf{X}_i^t will convergence to Pareto optimality \mathbf{X}^* .

Proof: Substituting (15) into (16), we obtain the following nonhomogeneous recurrence relation:

$$\begin{aligned} \mathbf{X}_i^{t+1} = & \left(1 - k_1 - k_2 + e^{-R(t+1)}\right) \cdot \mathbf{X}_i^t \\ & + k_1 \cdot \mathbf{X}_{\text{gbest},i}^t + k_2 \cdot \mathbf{X}_{\text{center}}^t \end{aligned} \quad (22)$$

where $i = 1, 2, \dots, N_P$, and (22) can be expressed as the matrix form

$$\begin{bmatrix} \mathbf{X}_i^{t+1} \\ \mathbf{X}_i^t \\ 1 \end{bmatrix} = \phi(t) \begin{bmatrix} \mathbf{X}_i^{t+1} \\ \mathbf{X}_i^t \\ 1 \end{bmatrix} \quad (23)$$

where the coefficient $\phi(t)$ is

$$\phi(t) = \begin{bmatrix} 1 - k_1 - k_2 + e^{-R(t+1)} & -e^{-R(t+1)} & k_1 \cdot \mathbf{X}_{\text{gbest},i}^t + k_2 \cdot \mathbf{X}_{\text{center}}^t \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

Then, assume that $e^{-R(t+1)} = r$, $k_1 + k_2 = K$, the characteristic polynomial of matrix $\phi(t)$ is obtained by (24):

$$\phi(t) - \lambda I = \begin{bmatrix} 1 - K - r & -r & k_1 \cdot \mathbf{X}_{\text{gbest},i}^t + k_2 \cdot \mathbf{X}_{\text{center}}^t \\ 1 & -\lambda & 0 \\ 0 & 0 & 1 - \lambda \end{bmatrix} \quad (25)$$

$$(1 - \lambda) \left[\lambda^2 - (1 - K + r) \cdot \lambda + r \right] = 0 \quad (26)$$

and the eigenvalues of $\phi(t)$ are

$$\lambda_1 = 1; \lambda_{2,3} = \frac{(1 - K + r) \pm \sqrt{(1 - K + r)^2 - 4r}}{2}. \quad (27)$$

Thus, the pigeon position can be stated as

$$\mathbf{X}_i^t = \gamma_1 \cdot \lambda_1 + \gamma_2 \cdot \lambda_2^t + \gamma_3 \cdot \lambda_3^t \quad (28)$$

where γ_1 , γ_2 , and γ_3 are constants.

Obviously, the eigenvalues are affected by the parameter K and r . The convergence condition of CMMOPIO algorithm is $\max\{|\lambda_2|, |\lambda_3|\} < 1$. That is,

$$\left| \frac{(1 - K + r) \pm \sqrt{(1 - K + r)^2 - 4r}}{2} \right| < 1. \quad (29)$$

Based on Assumption 3 and (17), the parameters k_1 and k_2 satisfy the following condition:

$$\begin{cases} 0 < r < 1 \\ 0 < K < 2(1 + r). \end{cases} \quad (30)$$

Thus, the condition needed to obtain the conclusion is discussed in the following cases.

Case 1 $(1 - K + r)^2 - 4r < 0$: In case 1, both of the eigenvalues λ_2 and λ_3 are complex numbers, thus

$$|\lambda_2|^2 = |\lambda_3|^2 = \frac{1}{4} \left| (1 - K + r)^2 - (1 - K + r)^2 + 4r \right| = r. \quad (31)$$

In fact, $\max\{|\lambda_2|, |\lambda_3|\} < 1$ requires only $r < 1$ in case 1. Also, the conditions of case 1 requires $-2\sqrt{r} + r + 1 < K < 2\sqrt{r} + r + 1$, and $r > 0$. Therefore, the convergence conditions of case 1 are

$$\begin{cases} 0 < r < 1 \\ -2\sqrt{r} + r + 1 < K < 2\sqrt{r} + r + 1. \end{cases} \quad (32)$$

Case 2 $(1 - K + r)^2 - 4r \geq 0$: In case 2, both of the eigenvalues λ_2 and λ_3 are real numbers, thus the condition of case 2 are equal to $r \geq 0$, and $K \leq -2\sqrt{r} + r + 1$ or $K \geq 2\sqrt{r} + r + 1$.

1) If $K \leq -2\sqrt{r} + r + 1$, then $\max\{|\lambda_2|, |\lambda_3|\} < 1$ requires only

$$|\lambda_2| = \left| \frac{(1 - K + r) + \sqrt{(1 - K + r)^2 - 4r}}{2} \right| < 1. \quad (33)$$

It leads to $0 < K \leq -2\sqrt{r} + r + 1$, $r < 1$.

2) If $K \geq 2\sqrt{r} + r + 1$, then $\max\{|\lambda_2|, |\lambda_3|\} < 1$ requires

$$|\lambda_3| = \left| \frac{(1 - K + r) - \sqrt{(1 - K + r)^2 - 4r}}{2} \right| < 1. \quad (34)$$

That is

$$\frac{(1 - K + r) - \sqrt{(1 - K + r)^2 - 4r}}{2} > -1. \quad (35)$$

It leads to $2\sqrt{r} + r + 1 \leq K < 2(1 + r)$, $r < 1$. Therefore, the convergence conditions of case 2 are

$$\begin{cases} 0 < r < 1 \\ 2\sqrt{r} + r + 1 \leq K < 2(1 + r). \end{cases} \quad (36)$$

Therefore, combining the two cases, (30) satisfies the convergence conditions of CMMOPIO algorithm.

According to (28), the convergence of the pigeon position can be calculated as

$$\lim_{t \rightarrow \infty} \mathbf{X}_i^t = \gamma_1. \quad (37)$$

When the $t \rightarrow \infty$ in (22), the value of the pigeon position can be calculated as

$$\lim_{t \rightarrow \infty} \mathbf{X}_i^t = \lim_{t \rightarrow \infty} (k_1 \cdot \mathbf{X}_{\text{gbest},i}^t + k_2 \cdot \mathbf{X}_{\text{center}}^t) / (k_1 + k_2). \quad (38)$$

Suppose that Assumptions 1 and 2 are valid, one has

$$\mathbf{X}_{\text{gbest},i}^{t-1} > \mathbf{X}_{\text{gbest},i}^t \text{ or } \mathbf{X}_{\text{gbest},i}^{t-1} < \mathbf{X}_{\text{gbest},i}^t \quad (39)$$

$$\mathbf{X}_{\text{center}}^t < \mathbf{X}_{\text{gbest},i}^t \text{ or } \mathbf{X}_{\text{center}}^t > \mathbf{X}_{\text{gbest},i}^t. \quad (40)$$

For CMMOPIO algorithm, the pigeon's best position $\mathbf{X}_{\text{gbest},i}$ will be further updated by the mutant mechanism which is based on the nondominated solutions which are close to the PFs. Thus

$$\lim_{t \rightarrow \infty} \mathbf{X}_{\text{gbest},i}^t = \mathbf{X}^*. \quad (41)$$

Moreover, the best position $\mathbf{X}_{\text{center}}$ is mutated by the pigeons' positions chosen from the current nondominated set $\mathbf{X}_{\text{gbest},i}$, $i = 1, 2, \dots, Np$. Thus, there exists

$$\lim_{t \rightarrow \infty} \mathbf{X}_{\text{center}}^t = \mathbf{X}^*. \quad (42)$$

Based on (41) and (42), (38) can be rewritten as

$$\lim_{t \rightarrow \infty} \mathbf{X}_i^t = \lim_{t \rightarrow \infty} (k_1 \cdot \mathbf{X}^* + k_2 \cdot \mathbf{X}^*) / (k_1 + k_2) = \mathbf{X}^*. \quad (43)$$

This proves Theorem 1 completely. ■

Theorem 2: When Assumption 1–3 are satisfied, the pigeons' velocity \mathbf{V}_i^t will convergence to 0.

Proof: According to (15), the velocity can be defined as

$$\mathbf{V}_i^{t+1} - (1 - K + r) \cdot \mathbf{V}_i^t + r \cdot \mathbf{V}_i^{t-1} = 0. \quad (44)$$

Then the characteristic polynomial of the coefficient matrix is

$$\left[\lambda^2 - (1 - K + r) \cdot \lambda + r \right] = 0 \quad (45)$$

and the eigenvalues of $\phi(t)$ are

$$\lambda_{45} = \frac{(1 - K + r) \pm \sqrt{(1 - K + r)^2 - 4r}}{2}. \quad (46)$$

Thus, the pigeon velocity can be stated as

$$\mathbf{V}_i^t = \gamma_4 \cdot \lambda_4^t + \gamma_5 \cdot \lambda_5^t \quad (47)$$

where γ_4 and γ_5 are constants.

Suppose Assumption 1–3 are satisfied, based on the analysis of Theorem 1, there exists

$$\lim_{t \rightarrow \infty} \mathbf{V}_i^t = 0. \quad (48)$$

Based on the convergence analysis, conclusions can be drawn that each pigeon in the population will converge to the best position with the iteration increasing. ■

V. EXPERIMENTAL WORK

A. Benchmark Functions

In order to verify the performance of our proposed CMMOPIO algorithm, some benchmark functions taken from specialized [29] are utilized. These benchmark problems are all multiobjective problems with two or three functions with maximum or minimum value.

B. Benchmark Functions Performance Measures

1) *Generational Distance*: The concept of the generational distance (GD) was presented by Veldhuizen and Lamont [30] as a way of testing how far the nondominated solutions found so far are from those in the Pareto optimal set and is defined as

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (49)$$

where n is the amount of the numbers in the set of the nondominated solutions found so far, and d_i is the Euclidean distance between each of these and the nearest elements in the Pareto optimal set. It is clear that these solutions are all contained in the Pareto optimal set when the GD is equal to zero. Thus, the values except zero will demonstrate the extend of how far the solutions from the PF of the function problem.

TABLE I
PARAMETERS OF THE CMMOPIO ALGORITHM

Description	Symbol	Value
weight factor in the map and compass operator	R	0.02
weight factor in the limit-cycle-based mechanism	λ	0.015
Population of the first repository	N	100
Population of the second repository	N_{rep}	100

2) *Spacing*: The concept of the spacing (SP) is introduced by Schott [31] which aims to measure the range variance of neighboring elements in the nondominated solutions found so far. It can be defined as

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^{i=n} (\bar{d} - d_i)^2} \quad (50)$$

where $d_i = \min_j (|f_1^i(\bar{x}) - f_1^j(\bar{x})| + |f_2^i(\bar{x}) - f_2^j(\bar{x})|)$, $i, j = 1, 2, \dots, n$, \bar{d} is the average value of all the d_i , and n is the number of the nondominated solutions found so far. When the value of SP is zero, it is indicated that all members in the PF currently available are equidistantly spaced.

3) *Inversed Generational Distance*: The concept of inversed generational distance (IGD) was proposed as a comprehensive index to evaluate the convergence performance and distribution performance of MOEAs [16]. The definition of IGD is as follows:

$$IGD(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|} \quad (51)$$

where P represents the points in true PF with the number $|P|$ and Q represents the Pareto optimal solution obtained by MOEAs. Thus, $d(v, Q)$ is the minimum Euclidean distance between each of individuals in P and the set Q . The smaller value of the IGD indicates the better performance of the algorithms.

C. Background of the Experiments

In order to verify the competence of our proposed algorithm, the contrast experiments with the representative of the state-of-the-art algorithms are conducted.

The experiments are conducted on the Lenovo Think-Vision and the operating system is Windows 7. All the algorithms are coded in MATLAB 2014a. The parameters of our proposed CMMOPIO algorithm are given in Table I.

D. Results and Analysis

1) *Performance Comparisons With Five Multiobjective Optimization Algorithms*: In this part, seven test functions are chosen from MOPs listed in Table II.

The comparative results of CMMOPIO with NSGA-II, SPEA2, MOPSO, MOEA-D, and CMOPIO are provided in Table III. The best, worst, mean, and standard deviation values of the performance measures are calculated to show the properties of the algorithms in more aspects.

Figs. 6–12 show the obtained nondominated solution points of six algorithms on seven benchmarks. As figures show,

TABLE II
BENCHMARK FUNCTIONS AND PARAMETERS

NO	Test Function	Search Space	Dimension
F1	$f_1 = x^2$ $f_2 = (x-2)^2$	$[-5, 5]$	1
F2	$f_1 = 1 - \exp(-\sum_{i=1}^{i=n} (x_i - \frac{1}{\sqrt{n}})^2)$ $f_2 = 1 - \exp(-\sum_{i=1}^{i=n} (x_i + \frac{1}{\sqrt{n}})^2)$	$[-4, 4]$	10
F3	$f_1 = \cos(\pi x_1 / 2) \cdot (1+g)$ $f_2 = \sin(\pi x_1 / 2) \cdot (1+g)$ $g = \sum_{i=2}^{i=n} (x_i - 0.5)^2$	$[0, 1]$	10
F4	$f_1 = x_1$ $f_2 = g \cdot h$ $g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i$ $h = 1 - \sqrt{f_1/g}$	$[0, 1]$	10
F5	$f_1 = x_1$ $f_2 = g \cdot h$ $g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i$ $h = 1 - (f_1/g)^2$	$[0, 1]$	10
F6	$f_1 = x_1$ $f_2 = g \cdot h$ $g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i$ $h = 1 - \sqrt{f_1/g} - (f_1/g) \cdot \sin(10\pi x_1)$	$[0, 1]$	10
F7	$f_1 = \cos(\pi x_1 / 2) \cdot \cos(\pi x_2 / 2) \cdot (1+g)$ $f_2 = \cos(\pi x_1 / 2) \cdot \sin(\pi x_2 / 2) \cdot (1+g)$ $f_3 = \sin(\pi x_1 / 2) \cdot (1+g)$ $g = \sum_{i=2}^{i=n} (x_i - 0.5)^2$	$[0, 1]$	10

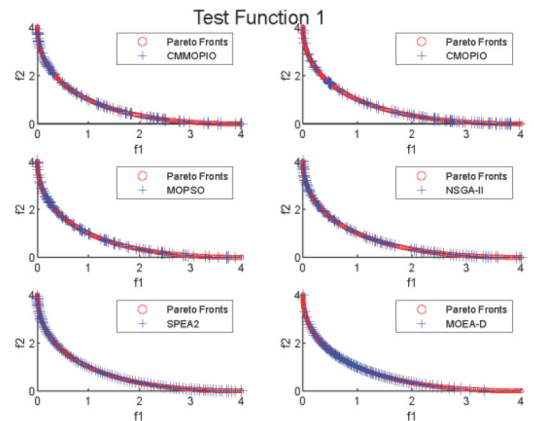


Fig. 6. Nondominated solution set of six algorithms on test function 1.

the Pareto optimal solutions of the proposed CMMOPIO algorithm could cover almost all of the true PF in seven benchmarks. Thus, the convergence of CMMOPIO algorithm is relative stable which is corresponding to the convergence analysis. From the above results in Table III, it is obvious

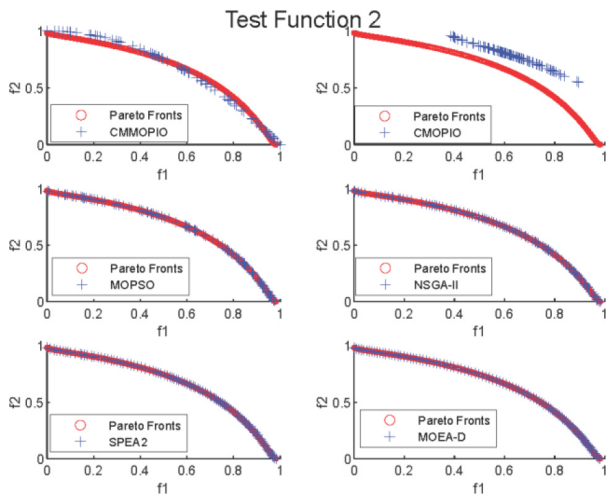


Fig. 7. Nondominated solution set of six algorithms on test function 2.

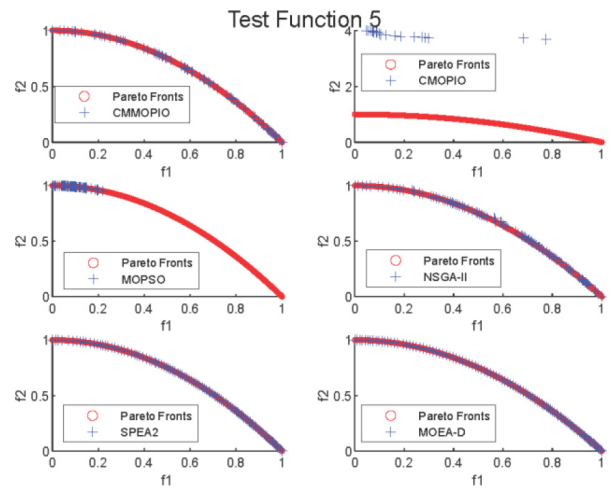


Fig. 10. Nondominated solution set of six algorithms on test function 5.

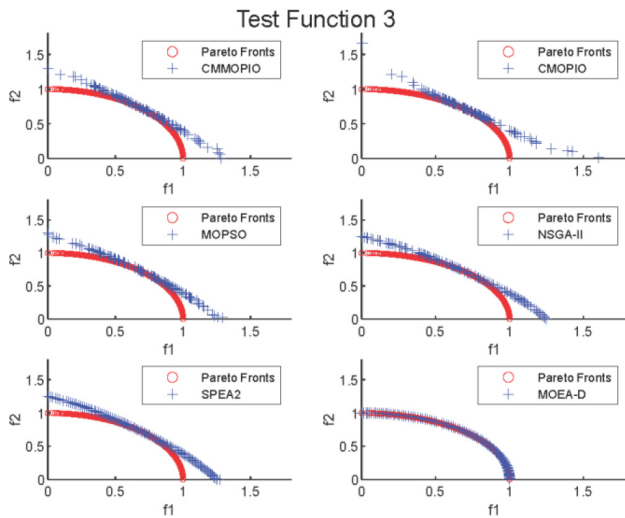


Fig. 8. Nondominated solution set of six algorithms on test function 3.

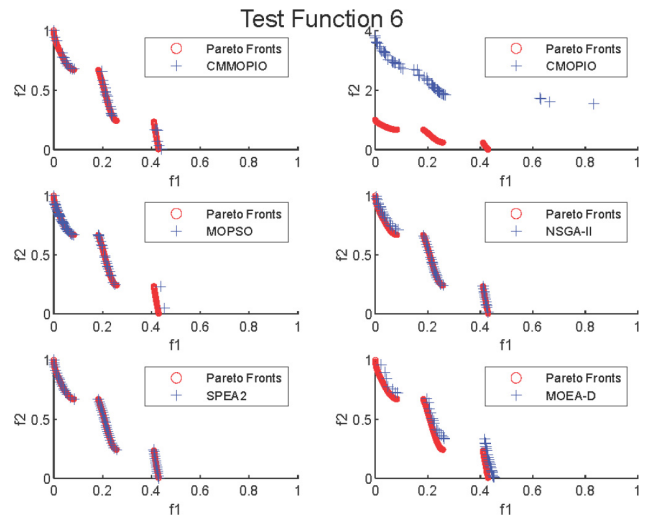


Fig. 11. Nondominated solution set of six algorithms on test function 6.

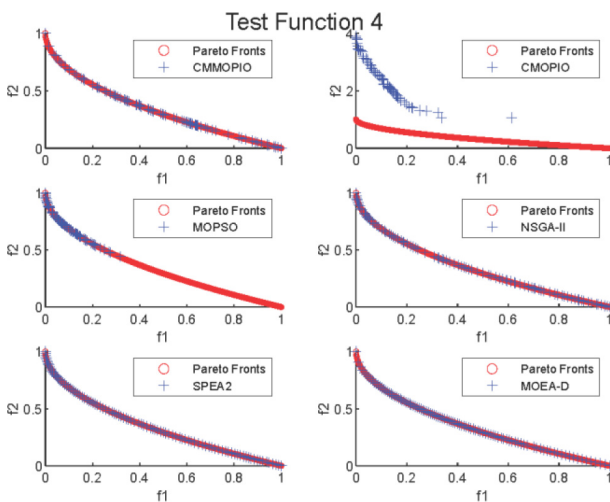


Fig. 9. Nondominated solution set of six algorithms on test function 4.

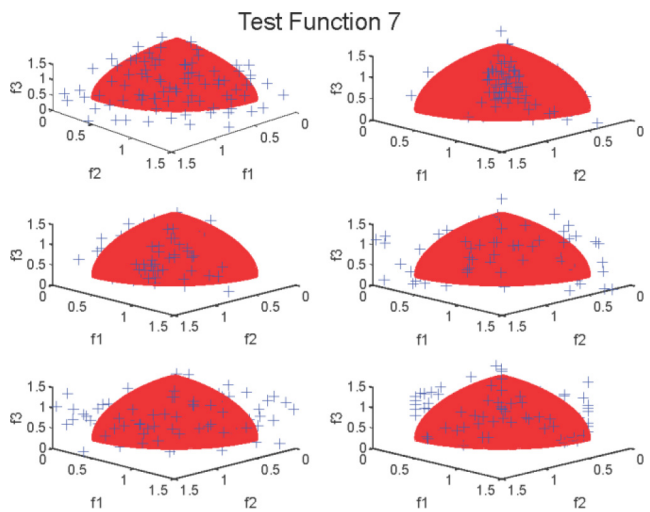


Fig. 12. Nondominated solution set of six algorithms on test function 7.

that CMMOPIO algorithm outperforms MOPSO, NSGA-II, SPEA2, MOEA/D, and CMOPIO algorithms on test function 1, 2, 4, 6, and 7 out of seven test functions in respect of the best

value of the performance measures. It is worse than NSGA-II and SPEA2 algorithms in respect of the average values of the performance values because of its bigger worst values.

TABLE III
RESULTS OF THE PERFORMANCE MEASURES FOR THE EIGHT TEST FUNCTIONS

			CMMOPIO	CMOPIO	MOPSO	NSGA-II	SPEA2	MOEA-D
F1	GD	best	1.5000e-09	0.0053	0.3880	5.7297e-04	6.0897e-04	5.4472e-04
		worst	0.0072	0.0053	1.2865	7.7186e-04	7.1417e-04	6.7503e-04
		average	0.0038	0.0053	0.7455	6.6471e-04	6.6539e-04	5.9392e-04
		Std.dev	0.0057	0	0.0584	1.7850e-04	7.6142e-05	9.5398e-05
	SP	best	2.1213e-09	0.0033	3.5530	0.0039	0.0042	0.0034
		worst	0.0078	0.0033	8.9611	0.0050	0.0049	0.0041
		average	0.0041	0.0033	6.2484	0.0045	0.0045	0.0037
		Std.dev	0.0057	0	0.2581	0.0010	5.0208e-04	5.5056e-04
F2	GD	best	2.0193e-11	8.3327e-04	0.0014	7.5740e-05	7.8111e-05	7.0936e-05
		worst	5.2630e-04	8.3327e-04	0.0018	1.0755e-04	9.2237e-05	8.7138e-05
		average	4.3325e-05	8.3327e-04	0.0016	9.2288e-05	8.5362e-05	7.9429e-05
		Std.dev	3.2917e-04	0	2.2206e-04	2.0692e-05	2.4135e-06	1.1419e-05
	SP	best	0	1.4119e-04	4.8857e-04	4.3747e-04	4.5779e-04	4.5093e-04
		worst	7.1367e-04	1.4119e-04	0.0084	5.5805e-04	5.6698e-04	5.5191e-04
		average	5.0391e-05	1.4119e-04	0.0059	4.9766e-04	5.0696e-04	4.9872e-04
		Std.dev	2.4794e-04	0	0.0062	5.2154e-05	5.9932e-05	6.5025e-05
F3	GD	best	1.5708e-04	0.4917	1.3300e-04	0.0067	0.0073	1.1876e-04
		worst	0.0704	0.5858	0.1579	0.0093	0.0081	0.0013
		average	0.0025	0.4976	0.0817	0.0082	0.0078	2.8405e-04
		Std.dev	0	0	0.0000	9.7456e-04	4.7662e-04	0.0012
	SP	best	1.6000e-05	0.6954	2.7861e-05	0.0604	0.0661	0.0010
		worst	0.0989	0.8284	0.2233	0.0810	0.0723	0.0116
		average	0.0034	0.7037	0.1140	0.0733	0.0698	0.0025
		Std.dev	0	0	0.0000	0.0089	0.0039	0.0106
F4	GD	best	7.3395e-05	1.4919	1.0680e-04	2.5694e-05	2.5949e-05	2.1248e-05
		worst	0.6412	1.5593	2.8820e-04	3.1770e-05	3.0410e-05	3.3462e-05
		average	0.1657	1.5026	1.9783e-04	2.8758e-05	2.8511e-05	2.8035e-05
		Std.dev	0.4325	0.0048	2.5728e-05	2.5477e-06	3.0145e-06	7.8311e-06
	SP	best	2.5885e-06	0.5413	1.9314e-05	1.3141e-04	1.3125e-04	1.1065e-04
		worst	0.9066	2.1167	2.0198e-04	1.5939e-04	1.5588e-04	1.8401e-04
		average	0.2341	1.2208	1.2264e-04	1.4623e-04	1.4272e-04	1.4193e-04
		Std.dev	0.6115	1.5754	1.3266e-04	7.9301e-06	1.1415e-05	3.9227e-05
F5	GD	best	0	2.9273e-04	2.0980e-04	2.5057e-05	1.9207e-16	2.9273e-04
		worst	1.1771	9.6986e-04	1.6129	3.1476e-05	3.5579e-05	9.6986e-04
		average	0.0417	8.8522e-04	0.5647	2.8581e-05	2.1618e-05	8.8522e-04
		Std.dev	1.1771	6.7713e-04	1.6127	2.2017e-06	1.5295e-11	6.7713e-04
	SP	best	0	3.3288e-04	2.7282e-05	1.2968e-04	1.8352e-15	3.3288e-04
		worst	1.6647	0.0011	2.2810	1.7905e-04	2.1943e-04	0.0011
		average	0.0589	0.0010	0.7984	1.4762e-04	1.1233e-04	0.0010
		Std.dev	1.6647	8.0429e-04	2.2809	2.4142e-05	1.5203e-10	8.0429e-04
F6	GD	best	0.0022	0.8127	9.0256e-05	3.4399e-05	1.4733e-05	4.5666e-05
		worst	0.7194	0.8127	0.0591	6.1931e-04	4.3729e-05	0.0021
		average	0.1574	0.8127	0.0255	2.0381e-04	2.0576e-05	3.0420e-04
		Std.dev	0.4812	0	0.0590	2.3607e-04	1.2421e-06	9.8577e-04
	SP	best	0.0030	1.1492	6.1350e-05	3.1276e-04	9.5622e-05	4.1411e-04
		worst	1.0492	1.1492	0.0364	0.0058	4.0530e-04	0.0181
		average	0.2212	1.1492	0.0157	0.0019	1.5835e-04	0.0027
		Std.dev	0.6804	0	0.0363	0.0023	2.0345e-05	0.0093
F7	GD	best	1.7411e-05	0.3464	2.3051e-05	0.0145	0.0104	2.6751e-04
		worst	0.6253	0.3911	0.2095	0.0234	0.0134	8.4967e-04
		average	0.1842	0.3721	0.0459	0.0192	0.0118	3.7416e-04
		Std.dev	0.1271	0.0447	0.1187	0.0089	0.0014	6.7713e-04
	SP	best	2.1035e-05	0.5999	2.4365e-05	0.1339	0.0975	3.3288e-04
		worst	1.0830	0.6766	0.3629	0.2050	0.1247	0.0082
		average	0.3186	0.6440	0.0795	0.1721	0.1099	0.0074
		Std.dev	0.2200	0.0766	0.2056	0.0711	0.0123	8.0150e-04

Particularly, CMMOPIO algorithm performs significantly better on test function 1, 4, 6, and 7. CMMOPIO algorithm can obtain the GD or SP result with an accuracy level of 10^{-6} , while other algorithms can only obtain the results under an accuracy of 10^{-5} or 10^{-4} . Therefore, CMMOPIO algorithms can always find a set of nondominated solutions closer to the true PFs in limited runs.

Through the contrast between CMMOPIO and CMOPIO algorithm, it can be seen that the nondominated solutions

obtained by the CMMOPIO algorithm are with good diversity and distributed uniformly. The distances between the points are approximate and the range of the points in the search space is wide. On the contrary, the CMOPIO algorithm can only find part of the solutions, which illustrates that CMOPIO algorithm is easy to be trapped into the local optimum. Therefore, it is obvious that the mutant mechanism plays a key role in searching the nondominated solutions.

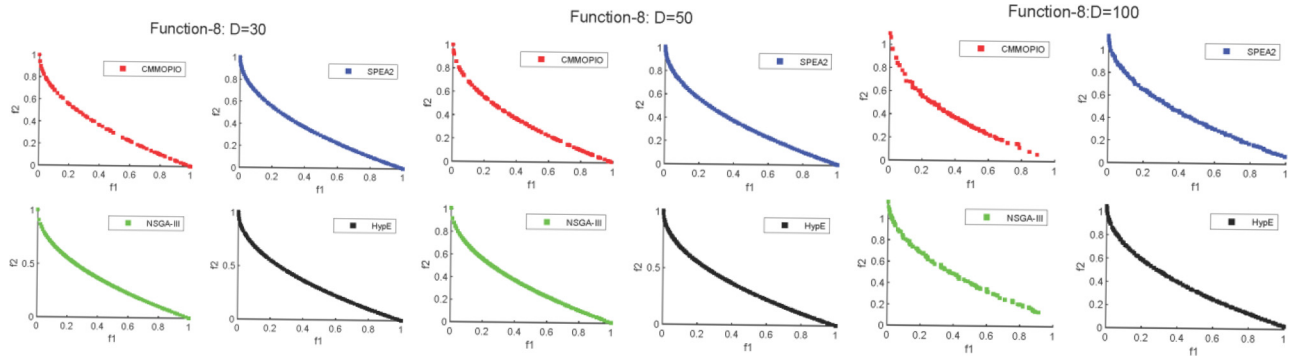


Fig. 13. Nondominated solution set on test function 8 in 30, 50, 100 dimensions.

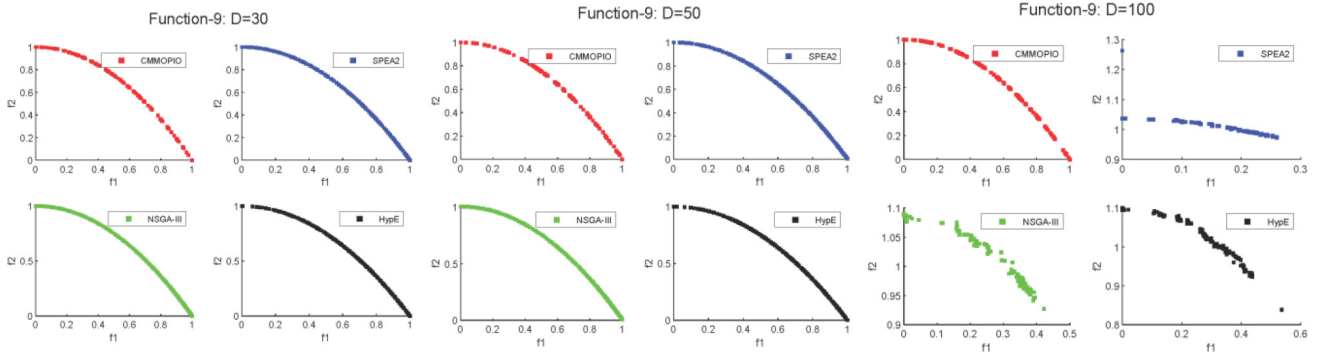


Fig. 14. Nondominated solution set on test function 9 in 30, 50, 100 dimensions.

2) *Performance Comparisons With CMMOPIO and Other Three Algorithms in High Dimensions:* To further investigate the effect of the proposed algorithm in high dimensions, three test functions with three dimensions are conducted between the CMMOPIO algorithm and SPEA2, NSGA-III, and HypE algorithms. The optimal Pareto solutions are drawn in curves. The convergence performance and distribution performance are estimated by indicator IGD which is depicted by histograms. The definition of test functions is as follows.

1) Test function 8 (minimizing MOP)

$$\begin{aligned}
 f_1 &= x_1 \\
 f_2 &= g(x) \left(1 - \sqrt{f_1/g} \right) \\
 g &= 1 + 9 \cdot \left(\sum_{i=2}^n x_i \right) / (n - 1). \tag{52}
 \end{aligned}$$

2) Test function 9 (maximizing MOP)

$$\begin{aligned}
 f_1 &= x_1 \\
 f_2 &= g(x) \left(1 - (f_1/g)^2 \right) \\
 g &= 1 + 9 \cdot \left(\sum_{i=2}^n x_i \right) / (n - 1). \tag{53}
 \end{aligned}$$

3) Test function 10 (minimizing MOP)

$$\begin{aligned}
 f_1 &= x_1 \\
 f_2 &= g(x) \left(1 - (f_1/g) - (f_1/g) \cdot \sin(10\pi x_1) \right) \\
 g &= 1 + 9 \cdot \left(\sum_{i=2}^n x_i \right) / (n - 1) \tag{54}
 \end{aligned}$$

where $x = (x_1, \dots, x_n)^T \in [0, 1]^n$.

It is obvious from Fig. 13 that all the four algorithms could find the optimal Pareto solutions with the dimension increases. In Fig. 14, CMMOPIO algorithm still keeps the stable search ability in three dimensions while the points of the other three algorithms gradually become spreading out at the high dimension. This phenomenon is even more obvious in Fig. 15 with segmented PFs. With the dimension increases, the numbers of PF segments found by the contrast algorithms reduce gradually. On the contrast, the proposed CMMOPIO algorithm could find five segments of PF until the dimension increases to 100. It can be seen more clearly in Fig. 16 that CMMOPIO algorithm have better performance in searching optimal solutions and convergence stability contrast to other three algorithms, especially, in high dimensions.

The conclusions can be drawn that the nondominated set of solutions in CMMOPIO algorithm distribute uniformly with good diversity and coverage of the true PF. The superior performance is mainly brought by utilizing the limit-cycle-based mechanism and mutant mechanism. The limit-cycle-based mechanism promotes the dynamic change of coefficients k_1 and k_2 to balance the global exploration and local exploitation which enhanced the ability of searching optimal pareto. The mutant mechanism helps the landmark operator migrated from SOPs still work well in MOPs. The mutant \mathbf{X}_{center}^l increases the range of searching step to avoid being trapped in the local optimal solutions which is easy to in face of during the last stage of optimization. What is more, the setting of coefficients k_1 and k_2 satisfies the convergence conditions in Section IV.

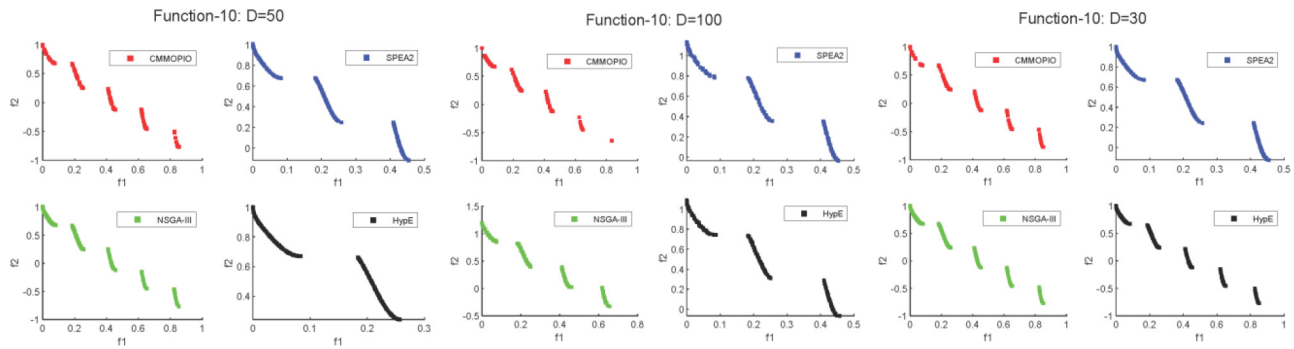


Fig. 15. Nondominated solution set on test function 10 in 30, 50, 100 dimensions.

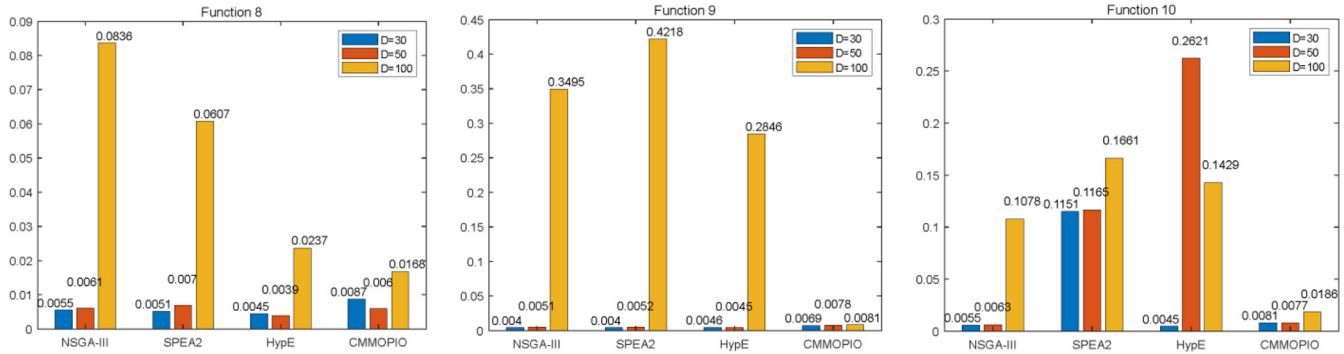


Fig. 16. IGD values of four algorithms in dimension 30, 50, 100 of function 8, 9, 10.

From the above experimental results, it is validated that our proposed CMMOPIO algorithm outperforms the other three algorithms in solving these three test function problems with high dimensions. It is obvious that the indicator IGD of CMMOPIO algorithm is relatively stable, and the mean data of IGD in CMMOPIO algorithm outperforms HypE, SPEA2, and NSGA-III algorithms in test functions 9 and 10. The results demonstrate the global exploration, local exploitation ability, and convergence stability of the CMMOPIO algorithm powerfully.

VI. CONCLUSION

In this article, a limit-cycle-based mutant multiobjective PIO was proposed. The limit-cycle-based mechanism integrates two independent cycles of basic PIO algorithm into one cycle to balance the global exploration and local exploitation. The proposed algorithm improves the exploratory capabilities of the MOPIO algorithm by introducing the mutant mechanism. Meanwhile, the convergence analysis and convergence conditions of the CMMOPIO algorithm are also investigated.

In the experiments, the proposed CMMOPIO algorithm is compared with several classical and state-of-the-art algorithms. The results of GD and SP reveal that the proposed algorithm is superior to other algorithms in finding the solutions approximate to the true PF. The results of IGD indicate the better searching ability and convergence stability of our proposed CMMOPIO algorithm. The contrast tests between the CMMOPIO algorithm and one version of the MOPIO algorithm has been carried out to prove the effect of mutant

mechanism. Thus, the proposed CMMOPIO algorithm can significantly improve the performance for solving MOPs.

Our future work will focus on further enhancing the performance of CMMOPIO algorithm, and extend it to solve more complicated MOPs with more optimized functions. Furthermore, the extension of CMMOPIO algorithm to solve some real-world problems will be studied in our future study.

REFERENCES

- [1] D. Martin, A. Rosete, J. Alcalá-Fdez, and H. Francisco, "A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 54–69, Feb. 2014.
- [2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001, pp. 487–410.
- [3] S. Y. Jiang and S. X. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
- [4] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 499–510, Feb. 2016.
- [5] D. He, L. Wang, and L. Yu, "Multi-objective nonlinear predictive control of process systems: A dual-mode tracking control approach," *J. Process Control*, vol. 25, pp. 142–151, Jan. 2015.
- [6] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.
- [7] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
- [8] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.
- [9] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

- [10] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength Pareto approach," *Swiss Federal Inst. Technol., Zürich, Switzerland, Rep.* 43, 1998.
- [11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [12] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [13] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [14] A. Muruganatham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016.
- [15] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *Swiss Federal Inst. Technol., Zürich, Switzerland, Rep.* 103, 2001.
- [16] H. Hanan, A. B. Azam, I. Amin, and R. Shahryar, "CGDE3: An efficient center-based algorithm for solving large-scale multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Wellington, New Zealand, 2019, pp. 350–358.
- [17] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
- [18] Q. Z. Lin and J. Y. Chen, "A novel micro-population immune multiobjective optimization algorithm," *Comput. Oper. Res.*, vol. 40, pp. 1590–1601, Jun. 2013.
- [19] Q. Z. Lin, Q. L. Zhu, P. Z. Huang, J. Y. Chen, Z. Ming, and J. P. Yu, "A novel hybrid multi-objective immune algorithm with adaptive differential evolution," *Comput. Oper. Res.*, vol. 62, pp. 95–111, Oct. 2015.
- [20] C. Dai, Y. P. Wang, and M. Ye, "A new multi-objective particle swarm optimization algorithm based on decomposition," *Inf. Sci.*, vol. 325, pp. 541–557, Dec. 2015.
- [21] Z.-H. Zhan, J. J. Li, J. N. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
- [22] H. B. Duan and P. X. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 24–37, 2014.
- [23] Y. Tian, X. Y. Zhang, C. Wang, and Y. C. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020, doi: [10.1109/TEVC.2019.2918140](https://doi.org/10.1109/TEVC.2019.2918140).
- [24] B. Zhang and H. B. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 14, no. 1, pp. 97–107, Jan./Feb. 2017.
- [25] Q. Xue and H. B. Duan, "Robust attitude control for reusable launch vehicles based on fractional calculus and pigeon-inspired optimization," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 89–97, Jan. 2017.
- [26] H. B. Duan and X. H. Wang, "Echo state networks with orthogonal pigeon-inspired optimization for image restoration," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2413–2425, Nov. 2016.
- [27] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.
- [28] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 49–172, Jun. 2000.
- [29] R. Ruiz-Cruz, E. N. Sanchez, F. Ornelas-Tellez, A. G. Loukianov, and R. G. Harley, "Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1698–1709, Dec. 2013.
- [30] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.*, vol. 8, no. 2, pp. 125–147, 2000.
- [31] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronautics and Astronautics, Massachusetts Inst. Technol., Cambridge, MA, USA, 1995.



Haibin Duan (Senior Member, IEEE) received the Ph.D. degree in control theory and engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005.

He was a Senior Visiting Scholar with the University of Suwon (USW), South Korea, in 2011. He is a Full Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, where he is the Vice Director of the State Key Laboratory of Virtual Reality Technology and Systems and the Head of the Bio-Inspired Autonomous Flight Systems Research Group. He is also a Visiting Research Fellow with Peng Cheng Laboratory, Shenzhen, China. He has authored or coauthored more than 70 publications. His current research interests are bio-inspired intelligence, and multi-UAV autonomous swarm control.

Prof. Duan received the National Science Fund for Distinguished Young Scholars of China in 2014. He is also enrolled in the Chang Jiang Scholars Program of China, Scientific and Technological Innovation Leading Talent of "Ten Thousand Plan"-National High Level Talents Special Support Plan, and Top-Notch Young Talents Program of China. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS.



Mengzhen Huo received the B.S. degree in automation science from Beihang University, Beijing, China, in 2018, where she is currently pursuing the Ph.D. degree with the BAFS Research Group, State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering.

Her current research interests are bio-inspired computation and unmanned aerial vehicle cooperative control.



Yuhui Shi (Fellow, IEEE) received the Ph.D. degree in electrical and electronic engineering from Southeast University, Dhaka, Bangladesh, in 1992.

He was with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China, from January 2008 to August 2017, the Electronic Data Systems Corporation, Indianapolis, IN, USA, from October 1998 to December 2007, and the Purdue School of Engineering and Technology, Indianapolis, from October 1995 to September 1998. He is a Chair

Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His main research interests are swarm intelligence and applications.

Mr. Shi is the Editor-in-Chief of the *International Journal of Swarm Intelligence Research* and an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.