IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 July 2023, accepted 19 July 2023, date of publication 24 July 2023, date of current version 31 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298058

RESEARCH ARTICLE

A Multi-Objective Optimization Approach Based on an Enhanced Particle Swarm Optimization Algorithm With Evolutionary Game Theory

KAIYANG YIN[1], BIWEI TANG[2], MING LI[3], AND HUANLI ZHAO[1]
[1]School of Electrical and Mechanical Engineering, Pingdingshan University, Pingdingshan, Henan 467036, China
[2]School of Automation, Wuhan University of Technology, Wuhan, Hubei 430070, China
[3]School of Economics and Management, Anhui Polytechnic University, Wuhu, Anhui 241000, China

Corresponding author: Ming Li (limingwhut@163.com)

This work was supported in part by the Project of the Science and Technology Department of Henan Province under Grant 222102220116, in part by the High-Level Talent Start-Up fund of Pingdingshan University under Grant PXY-BSQD-2021019, and in part by the Research Project of Higher Education in Henan Province under Grant 23A413008.

ABSTRACT Due to conflicts among objectives of multi-objective optimization (MO) problems, it remains challenging to gain high-quality Pareto fronts for different MO issues. Attempt to handle this challenge and obtain high-performance Pareto fronts, this paper proposes a novel MO optimizer via leveraging particle swarm optimization (PSO) with evolutionary game theory (EGT). Firstly, a modified self-adaptive PSO (MSAPSO) adopting a novel self-adaptive parameter adaption rule determined by the evolutionary strategy of EGT to tune the three key parameters of each particle is proposed in order to well balance the exploration and exploitation abilities of MSAPSO. Then, a parameter selection principle is provided to sufficiently guarantee convergence of MSAPSO followed after the analytical convergence investigation of this optimizer so as to assure convergence of the searched Pareto front toward the true Pareto front as far as possible. Subsequently, a MSAPSO-based MO optimizer is developed, in which an external archive is applied to preserve the searched non-dominated solutions and a circular sorting method is amalgamated with the elitist-saving method to update the external archive. Lastly, the performance of the proposed method is examined by 16 benchmark test functions against 4 well-known MOO methods. The simulation results reveal that the proposed method dominates its peers regarding the quality of the Pareto fronts for most of the studied benchmarks. Furthermore, the results of the non-parametric analysis confirm that the proposed method significantly outperforms its contenders at the confidential level of 95% over the 16 benchmarks.

INDEX TERMS Multi-objective optimization, particle swarm optimization, evolutionary game theory, convergence investigation, pareto front.

I. INTRODUCTION

Over the past two decades, multi-objective optimization (MO) has aroused increasing research interest due to its widespread real-world applications, such as energy dispatch [1], [2], job assembly [3] and controller optimization [4]. However, the conflicts among different objectives lead the issue of simultaneously gaining global optimum to each objective to be challenging or even impossible [5]. To handle this challenge, a MO problem is usually addressed to obtain a set of non-dominated solutions by considering trade-offs among different objectives [6]. Yet, since many MO problems are highly nonlinear both in objectives and constraints, gaining a series of non-dominated solutions with superior qualities still remains difficult [7].

Due to their swarm-based features and promising performances over nonlinear optimization problems, different evolutionary algorithms (EAs) have been developed within the last decade to handle MO issues [8], [9], [10]. Based on the typical decomposition multi-objective evolutionary algorithm (MOEA), Chen et al. have proposed a MO framework named MOEA/D-SCC by integrating the novel

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Augusto Ribeiro Chaves.

offspring selection process according to semi-supervised classification (SCC) [11]. To well balance the convergence and diversity, a novel MOEA/D method, named MOEA/D-AAP which applied an angle-based adaptive penalty (AAP) scheme for MOEA/D has been developed by Qiao et al. [12].

As one of the most remarkable EAs, particle swarm optimization (PSO) has been extensively used to handle different MO problems due to its easy implementation and promising convergence speed [13], [14]. To diversify the searched non-dominated solutions, an improved MOPSO approach (named MOPSOhv) adopting a mutation operator has been developed by García et al. [15]. According to R2-indicator and decomposition strategy, Li et al. have proposed a new PSO-based MO method called R2-MOPSO such that the diversification of the particle could be increased [16]. Similarly, aiming at sufficiently handling complex MO issues, a hybrid MOPSO based on R2 indicator has been established in Ref. [17]. For promoting the quality of the gained Pareto front, based on the MOEA/D and an adaptive replace strategy (ARS), an improved MOEA/D-ARS has been developed by Chen et al. [18]. For promoting the effectiveness of R2-MOPSO on large-scale MO issues, Li et al. have proposed R2-MOPSO-II by introducing a new bi-level archive maintaining strategy, a leader selection strategy, and a novel velocity updating strategy of the particle in R2-MOPSO [19]. Some other terrific studies using PSO-based MO algorithms to cope with different MO issues can be referred to Refs. [20], [21], [22], and [23].

Generally, at least two crucial issues need to be addressed in terms of applying PSO for MO problems. The first one is to trade off the exploration and exploitation abilities since the quality of the obtained front heavily relies on such two capabilities of the optimizer. However, it has been discovered that the standard PSO has difficulty in well balancing its exploration and exploitation, the Pareto front found by this algorithm could be a false one [24]. Due to the fact that the three main parameters (i.e., the inertia weight, the cognitive and social acceleration parameters) of the particle profoundly affect these two abilities of PSO, there have been studies concentrating on developing strategies to tune the three key parameters in PSO-based MO algorithms [25], [26], [27]. As different values of the three mentioned parameters determine the convergence property of PSO, it is vital to address and guarantee the convergence of PSO via renovating the three parameters to enhance the convergence of the non-dominated solution set toward the true Pareto front. This could be the second vital issue regarding applications of PSO on MO problems. Yet, the stochastic nature of PSO leads the theoretical convergence analysis pertaining to this optimizer to be difficult and challenging [28].

The aforementioned two deficiencies of PSO motivate this study under the background of using PSO to handle MO issues. The primary goal of this study is to efficiently handle MO issues and obtain high-quality Pareto fronts via remedying these two flaws of standard PSO noted above. To achieve this target, this paper firstly proposes a modified self-adaptive PSO (MSAPSO) by integrating standard PSO with evolutionary game theory (EGT). In the proposed MSAPSO, a novel self-adaptive parameter adaption strategy determined by the evolutionary strategy of EGT is presented to fine-tune the three key parameters of each particle in order to well balance the exploration and exploitation capabilities regarding MSAPSO. Afterward, followed by the analytical convergence investigation of MSAPSO, a parameter selection principle is provided to sufficiently guarantee the convergence of MSAPSO. Next, an MSAPSO-based MO method is established to handle different MO problems based on the developed MSAPSO. For obtaining a well-distributed Pareto front, an external archive is designed in the developed MSAPSO-based MO approach. Moreover, the circular sorting method [24] is combined with the elitist-saving method [29] to update the external achievement in the developed MO approach. The main potential contributions of this study could be summarized as follows:

(1) A novel self-adaptive parameter updating rule is developed in the proposed MSAPSO through implementing the evolutionary strategy of EGT, such that the exploration and exploitation powers of MSAPSO can be well balanced.

(2) A convergence-guaranteed parameter selection rule is proposed according to the analytical convergence investigation of MSAPSO in order to sufficiently assure the convergence of this developed optimizer.

(3) A MO approach is completed based on the developed MSAPSO, a size-fixed external archive designed beforehand, the circular sorting method and the elitist-saving method.

The performance of the proposed method is compared with those of 4 well-known MO algorithms over 16 benchmark test functions. The simulation results confirm that the proposed method outperforms its counterparts regarding the obtained Pareto fronts with respect to the majority of the studied benchmarks. Moreover, the results of the non-parametric statistical analysis also confirm that our proposed method significantly dominates its competitors at the confidential level of 95% over the 16 benchmarks.

The remaining of this study is organized as follows. The basic generality and formation of MO is given in Section II. Section III mainly introduces the proposed MSAPSO and the convergence-guaranteed parameter selection principle pertaining to this optimizer. The MSAPSO-based MO approach is stated in Section V. Section VI conducts the numerical simulations and result analysis. Section VI shows the parameter sensitivity study of MSAPSO. Section VIII completes this study by drawing conclusions and suggesting future works. The analytical convergence investigation of MSAPSO is shown in **Appendix A**.

II. THE BASIC FORMATION OF MO

It is well-known that handling a MO problem needs to optimize multiple conflicting objectives under several equality /inequality constraints. Thus, the general formation of a MO problem can be mathematically represented as follows [5]:

$$\text{Optimize}: F(x) = [F_1(x), F_2(x), \ldots, F_Q(x)] \quad (1)$$

$$\text{Subject to}: \begin{cases} h_i(x) = 0 & i = 1, 2, \ldots, cn_1 \quad (2) \\ g_j(x) < 0 & j = 1, 2, \ldots, cn_2 \quad (3) \\ x_k^l \leq x_k \leq x_k^u & k = 1, 2, \ldots, n \quad (4) \end{cases}$$

where F_d ($1 \leq d \leq Q$) represents the dth objective. Q is number of objectives needed to be optimized. $x = [x_1, x_2, \ldots, x_n]$ indicates the variable vector where n is the dimension of the vector. $h_i(x)$ and $g_j(x)$ denote the ith equality and jth inequality constraints, respectively. cn_1 and cn_2 are the total number of equality and inequality constraints, respectively. x_k^l and x_k^u are, respectively, the lower and upper boundaries of the kth optimization variable.

Note that since some basic definitions are used in our proposed MO method, they are recalled in the following contents(for minimizing a MO problem).

Definition 1 (Pareto Dominance): Solution S_a dominates solution S_b(denoted by $S_a \succ S_b$), if and only if $F_d(S_a) \leq F_{d_1}(S_b)$ for each $d \in [1, 2, \ldots, Q]$ and $F_d(S_a) < F_d(S_b)$, $\exists d \in [1, 2, \ldots, Q]$.

Definition 2 (Non-Dominated Solution): If there is no solution dominating solution S, it can be considered as a non-dominated one.

Definition 3 (Pareto Set): The set that composed by all non-dominated solutions.

Definition 4 (Pareto Front): The image of the Pareto set in the objective space.

III. INTRODUCTION OF STANDARD PSO

A. REVIEW OF STANDARD PSO

Inspired by birds flocking and homing, standard PSO was first proposed by Kennedy and Eberhart [30]. Each agent in PSO is regarded as a particle and represents a candidate solution for an optimization problem. During the search process, each particle dynamically updates its search information based on its own experience and those of its companions as follows [30]:

$$V_m^{k+1} = \omega V_m^k + c_1 r_1 (pbest_m^k - X_m^k) + c_2 r_2 (gbest - X_m^k) \quad (5)$$

$$X_m^{k+1} = X_m^k + V_m^{k+1} \quad (6)$$

where ω is the inertia weight. c_1 and c_2 represent the cognitive and social acceleration parameters of each particle, respectively. $pbest_m^k$ denotes the personal best position of the mth particle at kth iteration. $gbest$ indicates the global best position founded by the swarm. r_1 and r_2 are two stochastic numbers uniformly distributed in [0, 1]. Note that ω, c_1 and c_2 are predefined constants in real domain in standard PSO.

B. STATEMENTS OF BASIC PHILOSOPHIES FOR STANDARD PSO IMPROVEMENT

As a stochastic swarm-based algorithm, the standard PSO has been found to suffer from two typical drawbacks, namely, lower performance in trading-off its exploration and exploitation capabilities, as well as divergence toward the global optimum [31], [32]. These two flaws diminish its optimization performances over different MO issues. Thus, there exist strong necessities to surmount these deficiencies of standard PSO in terms of developing PSO-based MO optimizers.

It is notable that the exploration capability of PSO must be promoted in the early evolution to encourage particles to search through the entire solution space, so that the likelihood of missing the global optimum could be reduced possibly [31], [32], [33]. The exploitation ability needs to be intensified in the latter evolution in order to hearten particles to search carefully in a local area containing the global optimum, so that the possibility of finding the global optimum could be enhanced.

It is known that the three key control parameters of the particle significantly affect the exploration and exploitation powers of PSO. The basic philosophies with respect to influences of the three parameters on such two abilities can be summarized as follows [32]: (1) a large inertia weight benefits the exploration, whereas a small inertia weight consolidates the exploitation; (2) compared to the social acceleration parameter, a greater cognitive acceleration parameter fortifies the exploration ability; (3) compared to the cognitive acceleration parameter, a larger social acceleration parameter promotes the exploitation capability.

Evidently, the basic philosophies noted suggest a meaningful insight on improving PSO via dynamically tuning its three key parameters in a way of well balancing the exploration and exploitation powers. Apart from the impacts on the exploration and exploitation abilities, setting the three main aforementioned parameters also influences the convergence of PSO. In order to enhance the convergence of searched non-dominated solutions set toward the true Pareto front as far as possible, it remains paramount to analytically guarantee the convergence of PSO. However, the stochastic nature of PSO imposes difficulties in analytically investigating the convergence of PSO.

IV. STATEMENT OF THE PROPOSED MSAPSO

In order to promote the performances of PSO over different MO issues and gain high-quality Pareto fronts, this study concentrates its interest on developing a more advanced PSO-based MO algorithm by overcoming the two flaws of the standard PSO stated above. To this end, this study proposes a novel MSAPSO by leveraging standard PSO and EGT. In the developed MSAPSO, particles stick to the moving rule defined in standard PSO (referred to Eqs. (5)-(6)) to update their velocity and position information, respectively. Subsequently, a self-adaptive parameter updating principle is proposed in MSAPSO in order to well trade-off the exploration and exploitation abilities of this algorithm by implementing the evolutionary strategy of EGT. Afterward, a convergence-guaranteed parameter selection principle is provided for MSAPSO in order to assure the convergence

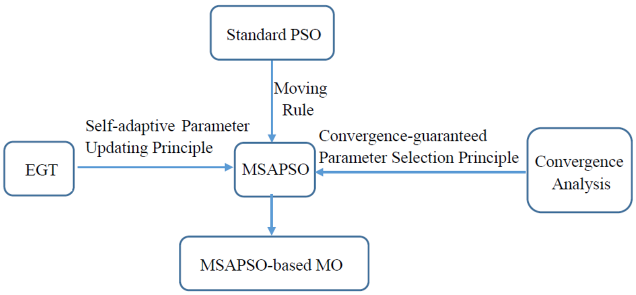


FIGURE 1. The main logical flow of the proposed MSAPSO.

of the found non-dominated solutions after the analytical convergence investigation of this developed algorithm. Finally, a MASPSO-based MO method is completed based on the proposed MSAPSO. The main logical flow of the proposed MSAPSO is visualized in Fig. 1. The self-adaptive parameter selection principle, the convergence investigation, the convergence-guaranteed parameter selection principle regarding MASPSO and the MSAPSO-based MO approach are detailed in the following contents.

A. THE SELF-ADAPTIVE PARAMETER UPDATING PRINCIPLE PROPOSED FOR MSAPSO

To well balance the exploration and exploitation capabilities of MSAPSO, following the philosophies described in Subsection III-B, we propose a self-adaptive parameter updating principle for this algorithm by applying the evolutionary strategy of EGT. Prior to introducing the proposed parameter updating principle, we first state the analogy between MSAPSO and EGT as follows: (1) each agent in EGT corresponds to a particle in MSAPSO; (2) each particle adopts three candidate strategies, namely, moving merely based on its inertia weight, personal best flight experience and the global best flight experience of the swarm, respectively; (3) the average performance gained by each particle under a given strategy among the three candidate strategies composes the payoff matrix of EGT.

Suppose that E_1, E_2 and E_3, respectively, denote the three candidate strategies noted above. Then, the payoff matrix used in our study can be computed as follows:

$$P_f = \begin{bmatrix} P(E_1) & \frac{P(E_1)-P(E_2)}{2} & \frac{P(E_1)-P(E_3)}{2} \\ \frac{P(E_2)-P(E_1)}{2} & P(E_2) & \frac{P(E_2)-P(E_3)}{2} \\ \frac{P(E_3)-P(E_1)}{2} & \frac{P(E_3)-P(E_2)}{2} & P(E_3) \end{bmatrix} \tag{7}$$

where P_f stands for the payoff matrix. $P(E_i)$ $(i = 1, 2, 3)$ indicates the payoff obtained by the particle via playing the ith strategy.

The replicator dynamic equation of EGT used in this paper is calculated by:

$$\dot{P}_i = -P_i(E_i \cdot P_f P^T - P P_f P^T) \tag{8}$$

where P_i $(i = 1, 2, 3)$ is the probability distribution of P_i over the candidate strategy E_i. P_f represents the payoff matrix. $P = [P_1, P_2, P_3]$ indicates the set of mixed strategies where we have that $\sum_{i=1}^{3} P_i = 1$ and $0 \leq P_i \leq 1$ $(i = 1, 2, 3)$

Assume that $E_s(k)$ denote the ratio of each candidate strategy in the case where the swarm converges to a stable position at iteration k. Then, $E_s(k)$ can be represented as follows:

$$E_s(k) = [Z_1(k), Z_2(k), Z_3(k)] \tag{9}$$

where we have that:

$$\sum_{i=1}^{3} Z_i(k) = 1 \tag{10}$$

Then, at each iteration k, the value of $P(E_i)$ in Eq. (7) can be gained according to the pervious flight experience of each particle in MSAPSO as:

$$P(E_i) = \frac{\sum_{k_1=1}^{k-1} Z_i(k_1) \sum_{l=1}^{Q} \mid F_l(x(k_1)) \mid}{k} \tag{11}$$

where k is the current iteration number of MSAPSO. $F_l(x(k_1))$ is fitness value of the lth objective of a MO issue obtained by the particle at iteration k_1. Q is the total number of objectives contained in a MO issue.

Once the value of $P(E_i)$ of each particle is generated by Eq. (11), it is subsequently used to fill the payoff matrix P_f based on Eq. (7). After obtaining the payoff matrix, the corresponding ratio of each candidate strategy, namely, $E_s(k) = [Z_1(k), Z_2(k), Z_3(k)]$, can be then computed by solving the replicator dynamic equation given by Eq. (8). Note that when solving Eq. (8) to gain $E_s(k)$, we have that $E_s(k) = P$ and $P_i = Z_i$ in this equation.

After obtaining the the three ratios $Z_1(k)$, $Z_2(k)$ and $Z_3(k)$ in $E_s(k)$ following the way described above, they are then implemented to renovate the three main parameters of each particle in the proposed self-adaptive parameter updating principle in MSAPSO as follows:

$$\omega^{k+1} = (\omega_s - \omega_f)\exp[-\frac{(\omega_s - \omega_f)k}{k_{max}\delta}] + \omega_f \tag{12}$$

$$c_1^{k+1} = (c_{1s} - c_{1f})\exp[-\frac{(c_{1s} - c_{1f})k}{k_{max}\delta}] + c_{1f} \tag{13}$$

$$c_2^{k+1} = (c_{2s} - c_{2f})\exp[\frac{(c_{2s} - c_{2f})k}{k_{max}\delta}] + c_{2f} \tag{14}$$

where:

$$\delta = \frac{Z_1(k) + Z_2(k)}{Z_3(k) + \triangle} \tag{15}$$

where subscripts "s" and "f" in each parameter are the initial and final values of the corresponding parameter, respectively. k_{max} is the given maximum iteration number. k is the current iteration number. $Z_i(k)$ $(i = 1, 2, 3)$ is the ratio of the ith strategy. $\triangle$ is a sufficiently small positive real number preventing the denominator in Eq. (15) becoming zero ($\triangle = 1e-04$). It notable that $\omega_s > \omega_f$, $c_{1s} > c_{1f}$ and $c_{2f} > c_{2s}$ in the above updating principle. Also, note that particles in MSAPSO adopt the moving rules defined by Eqs. (5)-(6) to renovate their velocity and position information, respectively.

It is important to note that the three ratios $Z_1(k)$, $Z_2(k)$, and $Z_3(k)$ in $E_s(k)$ represent a stable search direction of the swarm. This hints that the three ratios potentially denote the search stability nature of EGT. Thus, when these ratios are implemented in the above parameter updating principle, particles in MSAPSO could not only adapt the shape of the search space to optimize the search direction of the swarm but also may face the potential irregularity of the search space to avoid some local optimums as far as possible. This may imply that the applications of the three ratios in the developed parameter updating principle could enhance the performance of MSAPSO for solving MO issues.

Moreover, since PSO is a stochastic algorithm, the search behavior of each particle presents a nonlinear manner. For pandering to such search behavior, three key control parameters of particles are nonlinearly adjusted in the updating principle defined by Eqs. (12)-(15). Also, as the exponential function is known for its fast-growing nature, the three parameters of each particle are exponentially updated. This may promote the convergence speed of the proposed MSAPSO.

B. PARAMETER ANALYSIS OF MSAPSO

From Eqs. (12)-(14), one can infer that ω and c_1 decrease, whereas c_2 increases with the iteration number k increasing. Thus, based on the basic philosophies summarized in Subsection III-B, the exploration ability of MSAPSO is likely to be more preserved in the early phase of the evolution and would be taken over by the exploitation ability in the latter evolution.

Apart from the iteration number k, the trade-offs between the exploration and exploitation abilities can be also adjusted based on the value of δ. It is clear from Eq. (15) that a greater δ indicates a larger value of $(Z_1(k)+Z_2(k))$. A bigger value of $(Z_1(k) + Z_2(k))$ hints that the search direction of the swarm is more stable in the case where the particle mainly adopts the strategy of following its inertia and personal best flight memory. In such a case, it is desirable to increase the inertia weight and cognitive acceleration parameter to intensify the exploration ability of MSAPSO. Contrarily, a smaller value of δ represents a bigger value of $Z_3(k)$, which indicates that the search direction of the swarm could be more stable when the particle mainly plays the strategy of following the social flight experience of the swarm. In this case, it is meaningful to increase the social acceleration parameter of the particle to strengthen the exploitation capability.

Briefly, through employing the proposed self-adaptive parameter updating principle, the three control parameters of particles in MSAPSO can be adjusted in a way of complying with the basic philosophies of PSO development summarized in Subsection III-B. Thus, the proposed PSO algorithm could improve its performance in solving the MO issue.

C. CONVERGENCE-GUARANTEED PARAMETER SELECTION PRINCIPLE FOR MSAPSO

Since the convergence property of MSAPSO significantly affects its ability of finding high-quality non-dominated solutions, it is essential to analytically investigate and guarantee the convergence of this optimizer. To this end, we have theoretically analyzed the convergence of MSAPSO with respect to different values of the three main control parameters of particles in **Appendix A**.

One can readily obtain from **Appendix A** that MSAPSO converges, if and only if:

$$\begin{cases} 0 < c_1 r_1 + c_2 r_2 < 2\omega + 2 \\ -1 < \omega < 1 \end{cases} \tag{16}$$

where ω denotes the inertia weight of the particle. c_1 is the cognitive acceleration parameter of the particle. c_2 stands for the social acceleration parameter of the particle. r_1 and r_2 are two random numbers uniformly distributed in the range of [0,1].

Note that Eq. (16) is the necessary and sufficient condition for the convergence of MSAPSO. Despite obtaining this convergence condition, it still remains unknown how to set the initial and final values of the three parameters of the particle to sufficiently guarantee the convergence of MSAPSO merely from this condition. Herein, this subsection provides a convergence-guaranteed selection principle for the proposed algorithm by simultaneously considering the proposed self-adaptive parameter updating principle defined by Eqs. (12)-(15) and the convergence condition given by Eq. (16).

Lemma 1: The convergence of MSAPSO can be sufficiently guaranteed, only if the initial and final values of the three control parameters meet:

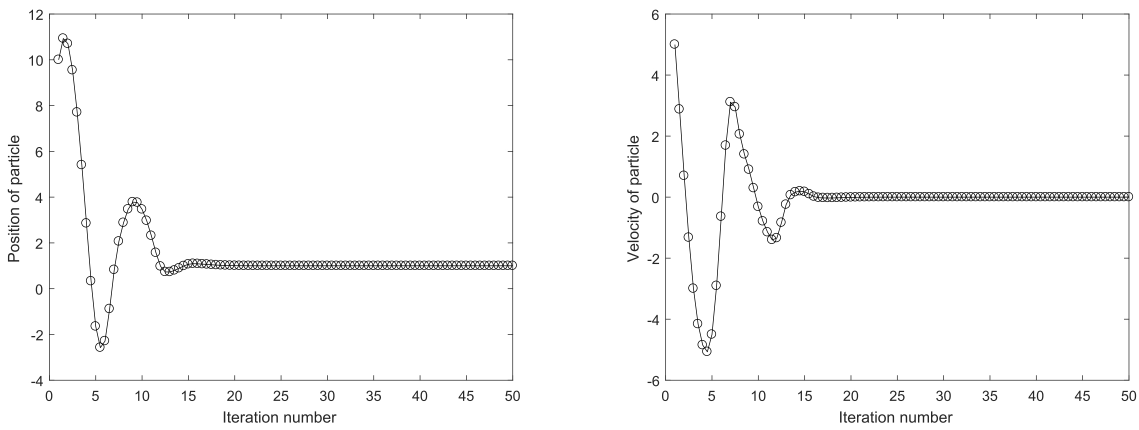
$$\begin{cases} c_{1s} + c_{1f} < 2\omega_{min} + 2 \\ -1 < \omega_{min} < \omega_{max} < 1 \\ 0 < c_{1f} = c_{2s} < c_{1s} = c_{2f} \end{cases} \tag{17}$$

Proof: It is trivial from Eqs. (13)-(14) that $c_1 + c_2 = c_{1s} + c_{1f}$ for each particle at any iteration for the cases where $c_{1s} = c_{2f}$ and $c_{1f} = c_{2s}$. Moreover, it is evident from Eqs. (12)-(14) that $\omega_{min} < \omega < \omega_{max}$, $c_{1f} < c_1 < c_{1s}$ and $c_{2s} < c_2 < c_{2f}$ in the proposed parameter updating principle. Since r_1 and r_2 are two random numbers in [0,1], one can readily obtain that:

$$\begin{cases} c_{1s} + c_{1f} < 2\omega_{min} + 2 \\ -1 < \omega_{min} < \omega_{max} < 1 \\ 0 < c_{1f} = c_{2s} < c_{1s} = c_{2f} \end{cases} \Rightarrow \begin{cases} 0 < c_1 r_1 + c_2 r_2 < 2\omega + 2 \\ -1 < \omega < 1 \end{cases} \tag{18}$$

Recall that the inequality on the right-hand side in Eq. (18) is the necessary and sufficient convergence condition for MSAPSO. Thus, the proof of **Lemma 1** can be easily held.

It is worth to noting that the initial and final values of ω, c_1 and c_2 are predefined constants in our proposed self-adaptive parameter updating principle. This implies that the



(a) Position (b) Velocity

FIGURE 2. **Convergence position and velocity trajectories of the particle in MSAPSO.**

sufficient convergence condition given by Eq. (17) can be easily satisfied via setting proper initial and final values of ω, c_1 and c_2. For sufficiently guaranteing the convergence of MSAPSO, we suggest that $\omega_{max} = 0.8$, $\omega_{min} = 0.4$, $c_{1s} = c_{2f} = 2$ and $c_{1f} = c_{2s} = 0.2$ based on the results of parameter sensitivity study as shown in the following Section VII.

Fig. 2 visualizes the convergent position and velocity trajectories of the particle under the above-suggested parameter selection. From this figure, it could be observed that the particle in the proposed algorithm illustrates a harmonic oscillation convergence behavior. This would imply that the exploration ability of this algorithm could be strengthened in the early stage of the evolution, whereas the exploitation capability dominates in the latter evolution.

V. MSAPSO-BASED MO METHOD

Several key technologies must be addressed in terms of applying PSO to solve a MO issue generally defined by Eqs. (1)-(4). The key technologies mainly include: (1) handling constraints of a MO problem; (2) updating the global and local personal best solutions of the particle; (3) preservations and renovations of non-dominated solutions searched by particles. The thereafter contents of this section first address these key issues in our MSAPSO-based MO method. Subsequently, the algorithmic steps of the this approach are stated at the end of this section.

A. HANDLING CONSTRAINTS OF A MO PROBLEM

Due to its easy implementation, the constraint handling method developed in Ref. [34] is first adopted in our developed MO approach to handle the equality and inequality constraints of a MO issue as defined by Eqs. (2)-(3) as follows:

$$VD_m = \sum_{i=1}^{cn_1} \mid h_i(x) \mid + \sum_{j=1}^{cn_2} \max\left(0, g_j(x)\right) \tag{19}$$

where VD_m denotes the violation degree of the mth particle in MSAPSO. The definitions of cn_1, $h_i(x)$, cn_2 and $g_j(x)$ can be referred to those given in Eqs. (2)-(3). $\mid h_i(x) \mid$ denotes the magnitude of $h_i(x)$.

Notably, the handling technology given by the above model is used to handle the equality and inequality constraints of a MO issue. For a MO issue unconstrained by any equality and inequality constraints, it is unnecessary to use this technology. After using Eq. (19) to calculate the violation degree of each candidate solution, the dominance-based rule [35] is then applied to select the non-dominated solution among two different candidate solutions in the developed MO method. The dominance-based rule can be described as follows: (1) the solution having a smaller violation degree dominates the one with a greater violation degree for any two different solutions owing different violation degrees; (2) the definition of Pareto dominance described in Section II is implemented to choose the non-dominated solution among two different candidate solutions having the same violation degrees.

From the above dominance-based rule, it can be seen that some infeasible solutions may allow us to enter into the next iteration. This could be interpreted by the fact that some infeasible solutions may contain valuable information about the solution space even if they may violate several constraints. When they are allowed to the next iteration, the diversification of the non-dominated solutions found by the swarm could be maintained as far as possible, which may thus enhance the possibility of finding superior-quality non-dominated solutions. Thanks to this advantage of the dominance-based rule, it is applied in our proposed MO method.

For handling the boundary constraints given by Eq. (4) of a MO problem, the following saturation mechanism is applied to modify any optimization variable x_j in our MO method as [36]:

$$x_j = \begin{cases} x_j^l & \text{if } x_j < x_j^l \\ x_j^u & \text{if } x_j > x_j^u \\ x_j & \text{otherwise} \end{cases} \tag{20}$$

where j is the number of variables to be optimized for a given MO issue. x_j^l and x_j^u are the lower and upper boundaries of x_j.

B. UPDATING THE PERSONAL AND GLOBAL BEST SOLUTIONS OF PARTICLES

Similar to some other PSO-based MO algorithms, the personal best solutions of particles in our MSAPSO-based MO method are updated based on the conventional method as follows [37]:

$$pbest_m^{k+1} = \begin{cases} X_m^k & \text{if } F(X_m^k) \succ F(pbest_m^k) \\ pbest_m^k & \text{otherwise} \end{cases} \tag{21}$$

where k is the current iteration number of the swarm. $pbest_m^{k+1}$ is the personal best solution of particle m at iteration $(k+1)$. $F(X_m^k)$ and $F(pbest_m^k)$, respectively, denote the fitness values of the current solution and personal best solution of particle m. "$\succ$" denotes the dominance operator using the dominance-based rule described in Subsection V-A.

Due to conflicts among different objectives, a set of non-dominated solutions are yielded in the MO problems. Referring to some currently-existing studies focusing on solving MO problems via PSO [37], this paper designs an external archive to preserve non-dominated solutions found by particles at each iteration. The global best solutions of particles are non-trivial from the designed archive based on the geographically-based strategy [24].

In the geographically-based strategy, the search objective explored by the current swarm is divided into different grids. Each divided grid contains several different non-dominated solutions. After selecting a grid using the roulette wheel by a density estimation operator, a non-dominated solution is randomly selected as the global best solution of the particle from this grid. Note that the number of non-dominated solutions involved in a grid determines the likelihood of this grid being selected. The more non-dominated solutions contained in a grid, the less likely the grid can be selected [24]. By using such a strategy, particles could be encouraged to search towards the less crowded solution spaces. This may decrease the possibility of missing some unknown solution spaces containing high-quality non-dominated solutions.

C. MAINTAIN OF THE EXTERNAL ARCHIVE

Similar to some other PSO-based MO algorithms, a fixed-size external archive is also designed in our MSAPSO-based MO method to iteratively save the non-dominated solutions found by the particle swarm [36]. At each iteration, the sorting method [29] is first used to renovate and check the allowance of entrance regarding each newly-searched solution to the external archive. After computing the values of objectives and violation degree of new solutions found by the swarm, the entrance of a newly-produced solution is allowed into the archive by the sorting method only if: (1) all non-dominates solutions in the archive cannot dominate the new solution; or (2) the new solution dominates any solution in the archive. Note that solutions dominated by the new solution need to be removed from the archive.

Because multiple non-dominated solutions could be simultaneously found by the swarm at each iteration, the size of the external archive increases explosively if all searched non-dominated solutions are preserved in the achieve, which not only leads to being computationally expensive to update the archive but also is not beneficial to balance the convergence and diversity of the swarm. To alleviate this problem, the elitist-preserving method [29] is implemented to prune the archive in the case where the size of the archive is full at each iteration. The algorithmic steps of the elitist-preserving method can be depicted as follows:

Step1: Sorting the non-dominated solutions saved in the archive in an ascending order according to the fitness values of the non-dominated solutions.

Step2: Computing the crowding distance of each non-dominated solution based on its fitness values.

Step3: Preserving N_a non-dominated solutions with the largest crowding distances in the archive where N_a is the predefined size of the archive.

Since the non-dominated solutions owing the greatest crowding distances are kept in the archive via the elitist-preserving approach, the preserved non-dominated solutions in the archive could be widely spread, which indicates that a well-distributed Pareto front with more diversifications could be gained by the elitist-preserving approach. After pruning the external archive following the sorting method and the elitist-preserving approach at each current iteration, the global best solution of the swarm for the next iteration is selected based on the geographically-based strategy depicted in Subsection V-B for the next iteration in order to well trade-off the exploration and exploitation of the swarm. The flow chart of the maintaining process concerning the external archive in our MSAPSO-based MO approach is displayed in Fig. 3. More detailed information on the sorting method and the elitist-preserving approach can be referred to Ref. [29].

D. ALGORITHMIC STEPS OF MSAPSO-BASED MO METHOD

The Algorithm 1 summarizes the algorithmic steps of the proposed MSAPSO-based MO approach. In this table, *NP*, k and k_m denote swarm size, the current iteration number and the maximum iteration number, respectively. Also, it is notable that the main loop the of the developed approach exists until the current iteration number reaches the given maximum iteration number.

VI. NUMERICAL SIMULATIONS

The proposed method is verified via 16 benchmark test functions (as depicted in Table 1) extracted from [5], [24], and [29]. The performance of the proposed method is compared with those of MMOABC [5], SAMOPSO [25], NSGA-II [29], and MOEA/D [38]. Attempt to conduct a

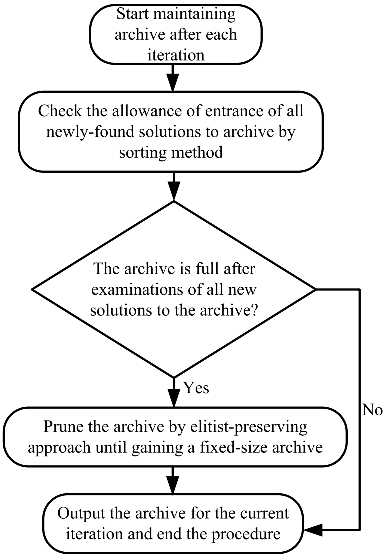


FIGURE 3. The maintaining process of the archive in MSAPSO-based MO method.

quantitative evaluation of the performance of each method, three commonly-used metrics described in Ref. [24] are adopted in this paper. The three metrics are the number of non-dominated solutions (NNS), generation distance (GD), and space metric (SM), respectively. Note that the quality of the Pareto front obtained by a given method is better with larger values of *NNS* and smaller values of *GD* and *SM*. For detailed ways of computing these metrics, the reader can be referred to [24].

For reducing the affects of randomness, we execute a Monte-Carlo test with 20 runs for each studied benchmark under a given method. In the Monte-Carlo experiment, the statistical results concerning each of the above metrics for each studied test function are reported and the average result with respect to each metric is examined. The swarm size and maximum iteration number of each method are set to 100 and 400, respectively. Moreover, the size of the external archive is set to 100 for each method in each run of the Monte-Caro experiment. The simulation parameters of our proposed method are set to be $\omega_{max} = 0.8$, $\omega_{min} = 0.4$, $c_{1s} = c_{2f} = 2$ and $c_{1f} = c_{2s} = 0.2$ based on the analysis results shown in Subsection IV-C. The simulation parameters of the four compared MO methods are referred to their original literature and shown in Table 2.

A. SIMULATION RESULTS OVER 16 TEST FUNCTIONS

After conducting the Monte-Carlo experiment for each considered MO method over each studied benchmark test function, the statistical results with respect to the aforementioned performance metrics are summarized in Tables 3-5. Note that the best average results gained concerning each metric are highlighted in boldface in these tables. The Pareto fronts searched by different methods for each test function are visualized in **Appendix B**.

Algorithm 1 The Algorithmic Steps of the MSAPSO-Based MO Method

Set needed simulation parameters and initialize the particle swarm
Compute the initial function values and violation degree of each particle
Set the initial solution of each particle as its *pbest* at the initial iteration
Renovate the external archive using circular sorting method at the initial iteration
if the size of the archive is full **do then**
  Remove some non-dominated solutions from the archive using the elitist-preserving approach
end if
while $k \leq k_{max}$ **do do**
  for $m = 1 : NP$ **do**
    Select *gbest* for particle *m* from the archive via the geographically-based method
    Update the velocity vector of particle *m* based on Eq. (5)
    Update position vector of particle *m* based on Eq. (6)
    Correct the position vector of particle *m* based on the saturation strategy given by Eq. (20)
    Calculate cost functions of particle *m*
    Compute the violation degree of particle *m* based on Eq. (19)
    Calculate the values of three ratios(i.e, $Z_1(k)$, $Z_2(k)$ and $Z_3(k)$) for particle *m* based on Eqs. (7)-(11)
    Update ω, c_1 and c_2 of particle *m* based on by Eqs. (12)-(15)
    Update *pbest* of particle *m* based on Eq. (21)
    Save *pbest* to the archive and renovate the archive by circular sorting method
  end for
  if the archive is full **do then**
    Remove some solutions from the archive using the elitist-preserving approach
  end if
  Increase the iteration number k by one
end while
Output non-dominated solutions saved in the archive

B. RESULT ANALYSIS

1) OVERALL ANALYSIS

It is clear from Tables 3-5 that all the five considered approaches cannot efficiently deal with $DTLZ_1$ and ZDT_4 probably due to complexities of objectives and multiple local optimums contained in these two benchmarks. This may indicate that despite being vital to balance the exploration and exploitation abilities of MO algorithms via parameter adaption strategies, it remains insufficient over some complicated MO issues with multiple local optimums. Combining parameter adaption strategy with some more

TABLE 1. Benchmark test functions.

Fun.	Variable dimension	Variable bounds	Objective Function
F_1	2	$[0.1,1]\cup[0,5]$	Function CONSTR in Ref. [29]
F_2	2	$[0,\pi]^n$	TNK Function in Ref. [29]
F_3	2	$[0.1,1]^n$	Test Function 4 in Ref. [24]
UF_1	30	$[0,1]\times[-1,1]^{n-1}$	Refer to Ref. [39]
UF_2	30	$[0,1]\times[-1,1]^{n-1}$	Refer to Ref. [39]
UF_4	30	$[0,1]^n$	Refer to Ref. [39]
CF_3	10	$[0,1]\times[-2,2]^{n-1}$	Refer to Ref. [39]
CF_4	10	$[0,1]\times[-2,2]^{n-1}$	Refer to Ref. [39]
ZDT_1	30	$[0,1]^n$	Refer to Ref. [5]
ZDT_3	30	$[0,1]^n$	Refer to Ref. [5]
ZDT_4	10	$[0,1]\times[-10,10]^{n-1}$	Refer to Ref. [5]
$DTLZ_1$	10	$[0,1]^n$	Refer to Ref. [29]
$DTLZ_2$	10	$[0,1]^n$	Refer to Ref. [29]
$DTLZ_3$	10	$[0,1]^n$	Refer to Ref. [29]
$DTLZ_4$	10	$[0,1]^n$	Refer to Ref. [29]
$DTLZ_5$	10	$[0,1]^n$	Refer to Ref. [29]

TABLE 2. Simulation parameters for compared methods.

Methods	Parameter setting
MMOABC	$L = 20$
NSGA-II	$P_c = 0.9$, $P_m = 0.1$
MOEA/D	$F = 0.5$, $P_c = 0.5$
SAMOSPO	$\omega_{max} = 0.9$, $\omega_{min} = 0.1$, $c_{1s} = c_{2f} = 2$, $c_{1f} = c_{2s} = 0.1$

advanced decomposition strategies, such as R2 indicator [19], could be a promising remedy to the issue noted above. Thus, we are considering the possibility of integrating our proposed MO method with some other outstanding decomposition strategies to further balance its convergence and diversity over more complicated MO issues in the coming future.

However, it is of great importance to note from Tables 3-5 that despite showing insufficient performances on $DTLZ_1$ and ZDT_4, the proposed method is still ranked the second and first among the five methods regarding the average *NNS*, *GD* and *SM* over these two functions, respectively. Moreover, apart from $DTLZ_1$, our proposed method shows the best performances over the rest 15 benchmarks in comparison with its peers. Thus, it allows us to generally conclude that the proposed method is highly competitive than its contenders for solving the 16 test functions.

It is worth noting that the above result analysis merely allows us to judge the average performance differences of the five methods over the 16 test functions. As shown in Tables 3-5, since the performances of different methods diversify from different test functions in terms of the mean *NNS*, *GD*, and *SM* metrics, a non-parametric analysis needs to be conducted to statistically investigate whether or not the five considered methods perform significantly different for solving the 16 benchmarks. To this end, we execute a non-parametric analysis and comparison in the following contents.

2) NON-PARAMETRIC ANALYSIS AND COMPARISON

It can be seen from Tables 3-5 that a better average *NNS* metric of a method for a test function corresponds to better average, *GD* and *SM* metrics of this method for this test function. Thus, only the *NNS* metric is regarded as an example in the conducted non-parametric analysis and comparison. Adopting the same analysis process, one can easily investigate the degrees of the five methods significantly differ from each other for the metrics of *GD* and *SM*.

In the non-parametric analysis and comparison, we first conduct a rank-based investigation to test the mean rank value of each considered approach over the 16 test functions. Afterwards, the non-parametric Friedman examination based

TABLE 3. Statistical results of *NNS* obtained by each method for each test function.

Functions	NNS	Methods				
		MSAPSO	SAMOPSO	MMOABC	MOEA/D	NSGA-II
F_1	Best	6.40E+01	5.20E+01	2.80E+01	2.10E+01	2.30E+01
	Average	**5.30E+01**	4.05E+01	2.08E+01	1.36E+01	1.17E+01
	Worst	4.30E+01	3.10E+01	1.30E+01	6.00E+00	8.00E+00
	Std. Dev.	5.25E+00	4.78E+00	4.79E+00	3.93E+00	3.56E+00
F_2	Best	2.20E+01	2.10E+01	1.50E+01	2.00E+01	1.90E+01
	Average	**1.46E+01**	1.37E+01	1.16E+01	1.28E+01	1.26E+01
	Worst	7.00E+00	7.00E+00	5.00E+00	6.00E+00	5.00E+00
	Std. Dev.	4.20E+00	3.33E+00	2.35E+00	3.35E+00	3.89E+00
F_3	Best	9.80E+01	8.90E+01	6.50E+01	3.90E+01	4.40E+01
	Average	**6.48E+01**	3.29E+01	1.37E+01	1.03E+01	9.60E+00
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	3.72E+01	3.29E+01	2.21E+01	1.29E+01	1.60E+01
UF_1	Best	2.90E+01	2.00E+01	1.00E+00	2.00E+00	1.00E+00
	Average	**1.25E+01**	1.23E+01	2.50E-01	6.00E-01	1.50E-01
	Worst	2.00E+00	3.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	8.25E+00	4.58E+00	4.44E-01	5.98E-01	3.66E-01
UF_2	Best	2.20E+01	2.00E+00	2.30R+01	2.40E+01	1.00E+00
	Average	**1.00E+01**	1.00E-01	7.15E+00	6.85E+00	5.00E-02
	Worst	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	6.04E+00	4.47E-01	6.53E+00	8.07E+00	2.24E-01
UF_4	Best	4.20E+01	3.70E+01	3.40E+01	1.70E+01	7.00E+00
	Average	**4.70E+00**	3.40E+00	2.25E+00	1.50E+00	5.50E-01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	1.19E+01	8.45E+00	7.55E+00	4.29E+00	1.76E+00
CF_3	Best	3.10E+01	5.00E+00	3.10E+01	4.10E+01	5.00E+00
	Average	**6.60E+00**	4.00E-01	4.25E+00	4.55E+00	6.50E-01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	7.84E+00	1.27E+00	9.19E+00	1.06E+01	1.31E+00
CF_4	Best	1.80E+01	1.50E+01	1.20E+01	1.80E+01	4.00E+00
	Average	**3.95E+00**	2.55E+00	2.80E+00	1.60E+00	4.00E-01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	4.57E+00	3.91E+00	3.02E+00	3.97E+00	9.95E-01
$ZDT1$	Best	9.50E+01	2.50E+01	1.70E+01	1.60E+01	2.90E+01
	Average	**7.70E+01**	3.85E+00	7.50E+00	3.50E+00	1.05E+01
	Worst	4.30E+01	0.00E+00	2.00E+00	1.00E+00	2.00E+00
	Std. Dev.	1.37E+01	7.31E+00	4.08E+00	3.38E+00	7.53E+00
ZDT_3	Best	6.70E+01	1.00E+01	1.80E+01	4.00E+00	3.10E+01
	Average	**4.58E+01**	1.65E+00	6.40E+00	1.25E+00	1.52E+01
	Worst	3.00E+01	0.00E+00	1.00E+00	0.00E+00	2.00E+00
	Std. Dev.	1.10E+01	3.00E+00	4.88E+00	9.10E-01	8.69E+00
ZDT_4	Best	3.20E+01	3.10E+01	7.00E+00	3.00E+00	2.00E+00
	Average	**1.47E+01**	5.65E+00	1.50E+00	1.32E+00	1.10E+00
	Worst	8.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Std. Dev.	5.40E+00	2.06E+01	1.47E+00	0.72E+00	3.08E-01
$DTLZ_1$	Best	5.00E+00	9.60E+01	2.00E+00	3.00E+00	1.00E+00
	Average	3.50E-01	**4.95E+00**	1.50E-01	2.21E-01	2.00E-01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	1.14E+00	2.14E+01	4.89E-01	6.96E-01	4.10E-01
$DTLZ_2$	Best	8.20E+01	2.70E+01	2.00E+00	5.60E+01	3.00E+01
	Average	**3.35E+01**	1.19E+01	7.00E-01	1.81E+01	8.55E+00
	Worst	1.20E+01	5.00E+00	0.00E+00	4.00E+00	1.00E+00
	Std. Dev.	1.84E+01	6.24E+00	8.01E-01	1.26E+01	7.58E+00
$DTLZ_3$	Best	2.20E+01	1.80E+01	1.00E+01	8.00E+00	1.00E+00
	Average	**3.85E+00**	3.40E+00	3.38E+00	2.73E+00	1.00E-01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	3.81E+00	4.83E+00	2.56E+00	2.43E+00	3.08E-01
$DTLZ_4$	Best	6.40E+01	4.00E+01	2.40E+01	2.00E+01	7.30E+01
	Average	**1.35E+01**	1.01E+01	9.25E+00	1.10E+00	1.09E+01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. Dev.	1.75E+01	1.23E+01	8.44E+00	4.46E+00	2.16E+01
$DTLZ_5$	Best	5.40E+01	2.90E+01	3.50E+01	4.10E+01	3.30E+01
	Average	**1.81E+01**	1.01E+01	5.05E+00	6.65E+00	1.23E+01
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.00E+00
	Std. Dev.	1.70E+01	7.17E+00	9.96E+00	1.20E+01	6.97E+00

TABLE 4. Statistical results of *GD* obtained by each method for each test function.

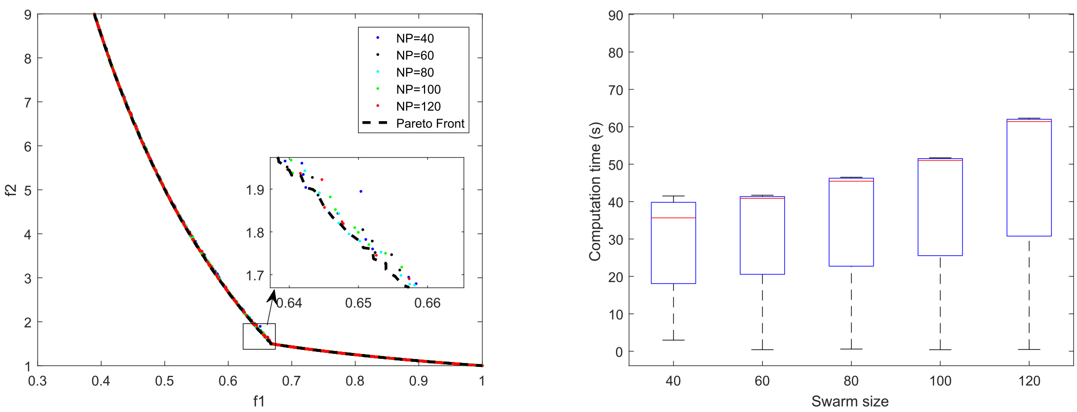
Functions	GD	Methods				
		MSAPSO	SAMOPSO	MMOABC	MOEA/D	NSGA-II
F_1	Best	3.67E-04	5.00E-04	4.97E-04	3.45E-04	5.13E-04
	Average	**5.41E-04**	6.59E-04	1.41E-03	1.55E-03	1.58E-03
	Worst	6.76E-04	9.06E-04	2.29E-03	2.68E-03	2.59E-03
	Std. Dev.	7.82E-05	1.03E-04	4.58E-04	6.94E-04	5.96E-04
F_2	Best	2.32E-04	2.40E-04	2.62E-04	2.51E-04	2.35E-04
	Average	**3.24E-04**	3.37E-04	3.54E-04	3.44E-04	3.51E-04
	Worst	4.36E-04	4.98E-04	5.60E-04	6.74E-04	4.22E-04
	Std. Dev.	4.97E-05	6.70E-05	7.02E-05	8.60E-05	5.03E-05
F_3	Best	2.85E-05	3.02E-05	1.04E-04	3.56E-04	2.81E-04
	Average	**2.87E-02**	3.57E-02	4.98E-02	2.45E-02	1.11E-01
	Worst	1.67E-01	1.42E-01	1.39E-01	1.34E-01	1.59E-01
	Std. Dev.	5.92E-02	5.75E-02	5.54E-02	4.55E-02	5.70E-02
UF_1	Best	1.76E-03	7.97E-04	7.29E-03	6.45E-03	7.29E-03
	Average	**6.45E-03**	5.46E-03	5.06E-02	5.65E-02	5.07E-02
	Worst	1.98E-02	4.85E-02	8.31E-02	1.11E-01	8.32E-02
	Std. Dev.	6.11E-03	1.06E-02	2.66E-02	2.93E-02	2.66E-02
UF_2	Best	1.13E-03	2.04E-03	1.49E-03	1.02E-03	1.94E-03
	Average	**1.86E-03**	3.54E-03	2.23E-03	2.57E-03	3.29E-03
	Worst	2.83E-03	4.52E-03	3.03E-03	3.37E-03	5.55E-03
	Std. Dev.	4.13E-04	7.23E-04	3.90E-04	5.61E-04	7.82E-04
UF_4	Best	9.75E-04	8.99E-04	1.54E-03	1.03E-03	1.56E-03
	Average	**2.08E-03**	1.88E-03	1.99E-03	1.91E-03	2.02E-03
	Worst	3.05E-03	2.91E-03	2.41E-03	2.62E-03	2.58E-03
	Std. Dev.	6.19E-04	5.37E-04	2.50E-04	3.80E-04	2.51E-04
CF_3	Best	1.47E-03	4.84E-03	3.77E-03	1.20E-03	7.20E-03
	Average	**3.01E-02**	1.08E-01	5.86E-02	7.48E-02	1.31E-01
	Worst	1.05E-01	4.40E-01	1.84E-01	2.37E-01	6.51E-01
	Std. Dev.	3.21E-02	1.13E-01	5.60E-02	7.75E-02	1.63E-01
CF_4	Best	4.03E-03	8.46E-03	9.73E-03	6.92E-03	1.10E-02
	Average	**2.54E-02**	3.32E-02	3.32E-02	2.82E-02	3.04E-02
	Worst	6.56E-02	1.63E-01	7.61E-02	6.63E-02	5.70E-02
	Std. Dev.	1.57E-02	3.23E-02	1.88E-02	1.67E-02	1.48E-02
$ZDT1$	Best	3.30E-08	1.11E-04	1.88E-04	1.17E-03	1.96E-04
	Average	**3.91E-05**	1.80E-02	7.45E-04	2.00E-02	5.10E-04
	Worst	1.55E-04	1.19E-01	1.47E-03	7.67E-02	1.34E-03
	Std. Dev.	4.14E-05	2.86E-02	3.94E-04	2.22E-02	2.72E-04
ZDT_3	Best	3.67E-05	5.29E-04	3.21E-04	4.18E-03	1.40E-04
	Average	**7.39E-05**	1.13E-02	1.79E-03	3.20E-02	5.04E-04
	Worst	1.13E-04	4.69E-02	7.88E-03	8.04E-02	1.16E-03
	Std. Dev.	2.47E-05	1.53E-02	1.75E-03	2.30E-02	3.21E-04
ZDT_4	Best	5.46E-03	5.59E-03	5.69E-03	7.78E-03	7.82E-03
	Average	**1.21E-02**	1.27E-02	1.31E-02	1.33E-02	1.39E-02
	Worst	1.58E-02	2.01E-02	1.82E-02	1.81E-02	2.15E-02
	Std. Dev.	2.85E-03	4.11E-03	3.72E-03	3.31E-03	3.63E-03
$DTLZ_1$	Best	7.64E-01	2.81E-02	9.85E-01	6.64E-01	1.96E+00
	Average	5.54E+00	**4.10E+00**	4.31E+00	6.97E+00	6.56E+00
	Worst	8.74E+00	1.40E+01	1.66E+01	2.09E+01	1.12E+01
	Std. Dev.	2.136E+00	4.28E+00	3.88E+00	4.59E+00	2.91E+00
$DTLZ_2$	Best	1.20E-03	5.83E-03	7.07E-03	1.23E-03	2.76E-03
	Average	**3.59E-03**	8.01E-03	1.82E-02	3.84E-03	6.50E-03
	Worst	6.97E-03	1.08E-02	3.49E-02	7.78E-03	1.17E-02
	Std. Dev.	1.32E-03	1.40E-03	7.80E-03	1.64E-03	2.78E-03
$DTLZ_3$	Best	1.85E-02	1.92E-02	4.23E-02	4.57E-02	8.68E-02
	Average	**5.16E-02**	5.48E-02	6.31E-02	8.80E-02	1.01E-01
	Worst	9.47E-02	7.75E-02	1.67E-01	1.67E-01	1.13E-01
	Std. Dev.	1.55E-02	1.58E-02	3.70E-02	3.70E-02	5.82E-03
$DTLZ_4$	Best	7.65E-04	3.05E-04	3.14E-04	1.15E-03	9.71E-04
	Average	**6.71E-03**	1.43E-02	2.06E-02	2.82E-02	7.45E-03
	Worst	3.06E-02	5.89E-02	1.26E-02	7.56E-02	6.95E-02
	Std. Dev.	8.04E-03	1.90E-02	3.81E-03	2.02E-02	1.64E-02
$DTLZ_5$	Best	1.84E-04	3.93E-04	1.37E-04	2.32E-04	4.23E-04
	Average	**5.68E-04**	7.27E-04	9.54E-04	8.34E-04	7.21E-04
	Worst	1.24E-03	1.37E-03	2.02E-03	3.99E-03	1.69E-03
	Std. Dev.	3.22E-04	2.25E-04	4.44E-04	9.82E-04	2.85E-04

TABLE 5. Statistical results of *SM* obtained by each method for each test function.

Functions	SM	Methods				
		MSAPSO	SAMOPSO	MMOABC	MOEA/D	NSGA-II
F_1	Best	2.94E-03	3.62E-03	1.89E-03	2.37E-03	2.23E-03
	Average	**4.56E-03**	4.98E-03	7.65E-03	5.68E-03	5.86E-03
	Worst	5.82E-03	6.93E-03	1.48E-02	1.09E-02	9.07E-03
	Std. Dev.	7.02E-04	8.02E-04	3.33E-03	2.06E-03	2.05E-03
F_2	Best	1.67E-03	1.75E-03	1.78E-03	1.80E-03	1.53E-03
	Average	**2.32E-03**	2.43E-03	2.61E-03	2.51E-03	2.59E-03
	Worst	3.53E-03	4.00E-03	4.91E-03	6.02E-03	3.12E-03
	Std. Dev.	4.83E-04	6.45E-04	7.58E-04	8.79E-04	4.27E-04
F_3	Best	2.84E-05	2.95E-04	9.45E-04	2.94E-03	2.38E-03
	Average	**2.42E-01**	2.93E-01	4.12E-01	1.87E-01	9.53E-01
	Worst	1.39E+00	1.21E+00	1.20E+00	1.18E+00	1.36E+00
	Std. Dev.	4.99E-01	4.99E-01	4.94E-01	4.05E-01	4.78E-01
UF_1	Best	1.41E-02	5.98E-02	3.59E-02	2.53E-02	2.08E-02
	Average	**4.67E-02**	5.24E-02	2.57E-01	2.61E-01	2.65E-01
	Worst	1.82E-01	3.95E-01	5.31E-01	4.41E-01	5.93E-01
	Std. Dev.	5.24E-02	8.74E-02	1.88E-01	1.45E-01	1.70E-01
UF_2	Best	9.39E-03	1.41E-02	1.28E-02	7.23E-03	1.53E-02
	Average	**1.59E-02**	2.76E-02	1.92E-02	2.22E-02	2.58E-02
	Worst	2.58E-02	3.95E-02	2.78E-02	2.98E-02	4.97E-02
	Std. Dev.	3.93E-03	7.01E-03	3.89E-03	5.31E-03	7.29E-03
UF_4	Best	4.02E-03	4.55E-03	3.65E-03	4.08E-03	4.24E-03
	Average	**5.85E-03**	6.32E-03	6.42E-03	6.15E-03	5.63E-03
	Worst	8.64E-03	8.32E-03	1.09E-02	8.64E-03	7.11E-03
	Std. Dev.	1.19E-03	1.08E-03	1.78E-03	1.10E-03	9.44E-04
CF_3	Best	7.27E-03	2.08E-02	3.00E-02	1.04E-02	2.44E-02
	Average	**1.58E-01**	4.61E-01	2.97E-01	4.15E-01	5.85E-01
	Worst	6.36E-01	1.79E+00	1.07E+00	1.50E+00	2.70E+00
	Std. Dev.	1.78E-01	4.77E-01	2.99E-01	5.03E-01	7.15E-01
CF_4	Best	2.43E-02	4.69E-02	4.75E-02	3.68E-02	3.51E-02
	Average	**1.33E-01**	1.90E-01	1.87E-01	1.19E-01	1.43E-01
	Worst	3.69E-01	1.01E+00	4.25E-01	2.62E-01	3.06E-01
	Std. Dev.	8.69E-02	2.04E-01	1.05E-01	6.93E-02	8.27E-02
$ZDT1$	Best	3.23E-07	6.52E-04	1.16E-03	4.91E-03	1.20E-03
	Average	**3.51E-04**	4.66E-02	3.83E-03	4.57E-02	2.82E-03
	Worst	1.38E-03	3.12E-01	6.70E-03	1.46E-01	5.79E-03
	Std. Dev.	3.66E-04	7.19E-02	1.71E-03	3.73E-02	1.09E-03
ZDT_3	Best	3.06E-04	3.73E-03	2.12E-03	1.22E-02	1.02E-03
	Average	**5.93E-04**	4.77E-02	7.87E-03	6.80E-02	2.83E-03
	Worst	1.02E-03	1.93E-01	2.85E-02	1.26E-01	6.28E-03
	Std. Dev.	1.93E-04	6.07E-02	6.33E-03	3.87E-02	1.44E-03
ZDT_4	Best	1.04E-02	3.02E-02	3.62E-02	3.46E-02	3.52E-02
	Average	**5.60E-02**	5.63E-02	5.88E-02	5.89E-02	6.98E-02
	Worst	7.76E-02	8.44E-02	8.77E-02	9.02E-02	9.14E-02
	Std. Dev.	1.57E-02	1.52E-02	1.82E-02	1.75E-02	1.68E-02
$DTLZ_1$	Best	2.44E+00	2.80E+00	1.93E+00	2.79E+00	4.54E+00
	Average	1.77E+01	**1.50E+01**	2.25E+01	2.45E+01	2.09E+01
	Worst	4.23E+01	8.14E+01	7.73E+01	7.87E+01	4.68E+01
	Std. Dev.	1.05E+01	1.88E+01	1.66E+01	2.25E+01	1.14E+01
$DTLZ_2$	Best	1.16E-02	3.36E-02	3.26E-02	1.06E-02	1.86E-02
	Average	**2.70E-02**	5.45E-02	6.75E-02	2.54E-02	3.28E-02
	Worst	6.10E-02	7.82E-02	1.09E-01	4.96E-02	4.86E-02
	Std. Dev.	1.20E-02	1.18E-02	2.27E-02	1.04E-02	9.90E-03
$DTLZ_3$	Best	1.04E-01	1.34E-01	1.51E-01	1.22E-01	1.51E-01
	Average	**1.92E-01**	1.93E-01	2.33E-01	2.81E-01	4.86E-01
	Worst	2.85E-01	2.45E-01	4.07E-01	5.39E-01	0.98E-01
	Std. Dev.	5.08E-02	2.76E-02	6.11E-02	1.04E-01	2.64E-01
$DTLZ_4$	Best	9.45E-03	9.98E-02	9.99E-02	3.01E-03	4.85E-03
	Average	**3.43E-02**	2.40E-01	5.40E-01	5.44E-02	5.31E-02
	Worst	1.19E-01	4.66E-01	4.75E-01	1.64E-01	1.42D-01
	Std. Dev.	2.92E-02	1.62E-01	1.73E-01	4.55E-02	4.00E-02
$DTLZ_5$	Best	1.63E-03	2.78E-03	8.98E-04	1.06E-03	3.17E-03
	Average	**4.30E-03**	5.26E-03	5.25E-03	4.24E-03	5.45E-03
	Worst	1.08E-02	1.03E-02	1.02E-02	2.32E-02	1.38E-02
	Std. Dev.	2.58E-03	1.93E-03	3.17E-03	5.22E-03	2.56E-03

on the average rank value of each method is executed followed by the pairwise post hoc Bonferroni-Dunn to evaluate the average performances of the five methods over the 16 test functions.

Table 6 summarizes the mean rank value of *NNS* obtained by each method over the 16 test functions based on the numerical results reported in Table 3. It is apparent from Table 6 that the proposed method is followed by SAMOPSO, MMOABC, MOEA/D, and NSGA-II in terms of the average *NNS* performance over the 16 test functions. However, the average rank analysis depicted here is insufficient to confirm the significant differences between the proposed



(a) Obtained Pareto fronts (b) Computation time

FIGURE 4. Obtained Pareto fronts and computation time for Function F_1 under different swarm size settings.

method and its four encounters. Attempting to examine the significant distinctions between the proposed method and the four compared methods, the statistical comparisons based on more advanced non-parametric analysis need to be executed. To this end, the Friedman test based on the average rank values summarized in Table 6 is implemented followed by the pairwise post hoc Bonferroni-Dunn test in the thereafter contents.

Since the performances of 5 approaches over 16 test functions are compared in this paper, the F-statistic value of the Friedman test with the confidential level of 95% is 2.52 based on Matlab command: $finv(aN-1, (N-1)(K-1))$, where a, N and K are the confidential levels, the total number of test methods and functions, respectively. According to the numerical results shown in Table 6, we can readily obtain that the Friedman statistic value is 18.86. Due to the fact that the Friedman statistic value is greater than the F-statistic value, the null hypothesis that each method performs equally for all test functions can be rejected. Thus, we can conclude that the five considered methods significantly differ from each other for solving the 16 benchmarks with regard to the mean *NNS* metric at the confidential level of 95%.

It is noticeable that the non-parametric Friedman test depicted above only allows us to infer that the five methods are significantly different over the 16 test functions at a confidential level of 95%, rather than that the proposed method significantly outperforms the rest 4 methods at the given confidential level. Herein, the pairwise post hoc Bonferroni-Dunn test is conducted to detect whether the proposed method significantly dominates the other four compared methods in terms of the average *NNS* performance over the 16 test functions at the given confidential level. Note that a given approach significantly outperforms another one at a given confidential level if the mean rank distinctions between this method and its competitor are no less than the critical difference value of the Bonferroni-Dunn test.

Recall that 5 methods are used to deal with 16 benchmarks in this paper. Therefore, it can be obtained that the critical difference value equals 1.5038. It can be observed from

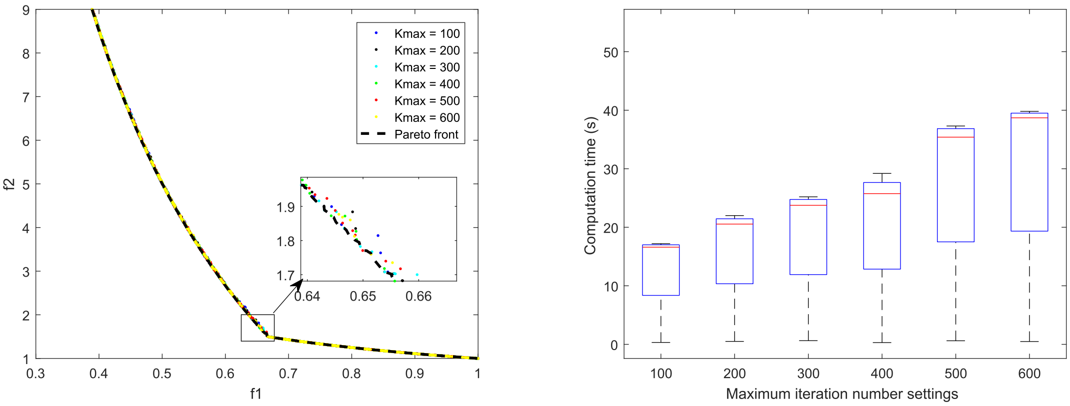
TABLE 6. Mean rank values of *NNS* gained by different methods for 16 test functions.

Functions	Methods				
	MSAPSO	SAMOPSO	MMOABC	MOEA/D	NSGA-II
F_1	1	2	3	4	5
F_2	1	2	5	3	4
F_3	1	2	3	4	5
UF_1	1	2	3	4	5
UF_2	1	4	2	3	5
UF_4	1	2	3	4	5
CF_3	1	5	3	2	4
CF_4	1	3	2	4	5
ZDT_1	1	4	3	5	2
ZDT_3	1	4	3	5	2
ZDT_4	1	2	3	4	5
$DTLZ_1$	2	1	5	3	4
$DTLZ_2$	1	3	5	2	4
$DTLZ_3$	1	2	3	4	5
$DTLZ_4$	1	3	4	5	2
$DTLZ_5$	1	3	5	4	2
Average	1.06	2.75	3.43	3.75	4.00

Table 6 that the distinctions of average rank value between MSAPSO, SAMOPSO, MMOABC, MOEA/D, and NSGA-II are 1.69, 2.37, 2.69, 2.94, respectively. Since these distinction values are all greater than the critical difference value ($C_d = 1.5038$). Consequently, it can be conclusive that the proposed method is significantly better than the four compared methods over the 16 test functions at the confidential level of 95%. This indicates that the proposed method can be regarded as an efficient alternative in the field of dealing with MO problems.

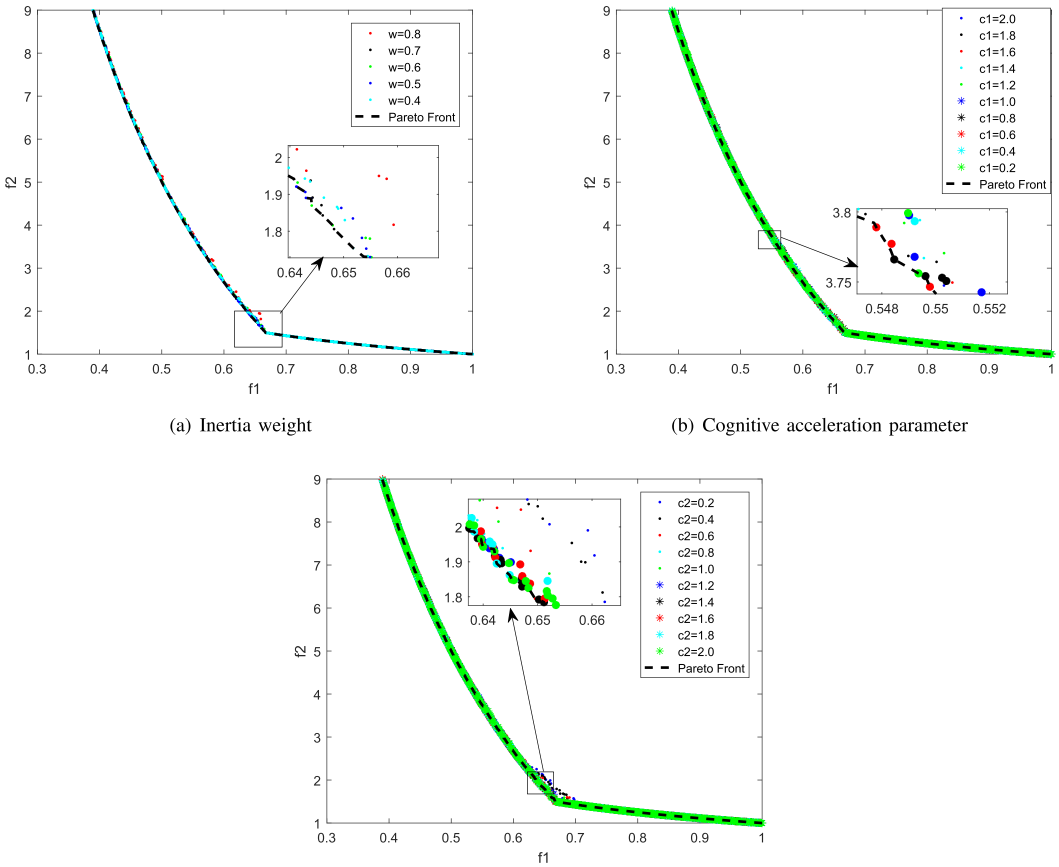
VII. PARAMETER SENSITIVITY STUDY OF MSAPSO

This section conducts a sensitivity investigation of different parameter settings regarding the swarm size, maximum iteration number, inertia weight, and cognitive and social acceleration parameters in MSAPSO. In the conducted sensitivity study, only Function F_1 is used as a pilot test function thanks to its simplicity. Moreover, the Monte-Carlo test stated in Section VI is implemented for each single sensitivity investigation. Besides, the remaining parameters



(a) Obtained Pareto fronts (b) Computation time

FIGURE 5. **Obtained Pareto fronts and computation time for Function F_1 under different maximum iteration number settings.**



(a) Inertia weight (b) Cognitive acceleration parameter

(c) Social acceleration parameter

FIGURE 6. **Obtained Pareto fronts for Function F_1 under different settings of the three control parameters.**

have remained as recommended in Section VI for the case where one given parameter is studied.

Also, the three performance metrics depicted in the above section are adopted in the sensitivity study. Note that since the swarm size and maximum iteration number also significantly affect the computation time (denoted by CT in this work) of the proposed optimizer, the computation time is used as an extra metric to study the sensitivities of these two parameters on the performance of our proposed method. The descriptions and numerical simulation results with respect to different parameter settings are detailed in the following contents.

A. SWARM SIZE

The swarm size (denoted by *NP*) is the total number of particles contained in MSAPSO, which profoundly

TABLE 7. Simulation results of different performance metrics for Function F_1 under different settings of swarm size in MSAPSO.

Items	Results	Swarm size				
		40	60	80	100	120
NNS	Best	3.90E+01	4.40E+01	5.20E+01	5.20E+01	5.30E+01
	Average	3.60E+01	3.62E+01	4.20E+01	4.25E+01	**4.28E+01**
	Worst	3.40E+01	2.80E+01	3.20E+01	3.20E+01	3.40E+01
	Std. Dev.	2.28E+00	5.98E+00	7.40E+00	6.86E+00	6.55E+00
GD	Best	3.44E-04	3.19E-04	3.32E-04	3.14E-04	2.58E-04
	Average	4.67E-04	4.45E-04	4.20E-04	4.02E-04	**3.57E-04**
	Worst	6.71E-04	6.10E-04	5.20E-04	5.26E-04	5.02E-04
	Std. Dev.	1.23E-04	1.05E-04	6.27E-05	7.51E-05	8.19E-05
SM	Best	2.58E-03	2.58E-03	2.41E-03	2.12E-03	3.28E-03
	Average	3.61E-03	3.46E-03	3.20E-03	2.79E-03	**2.73E-03**
	Worst	5.55E-03	4.79E-03	4.20E-03	4.06E-03	3.28E-03
	Std. Dev.	1.17E-03	8.00E-04	6.04E-04	6.82E-04	3.84E-04

TABLE 8. Simulation results of computation time (*CT* in second) for Function F_1 under different settings of swarm size in MSAPSO.

Item	Results	Swarm size				
		40	60	80	100	120
CT	Best	3.32E+01	4.07E+01	4.49E+01	5.07E+01	6.11E+01
	Average	**3.81E+01**	4.10E+01	4.60E+01	5.13E+01	6.17E+01
	Worst	4.15E+01	4.17E+01	4.65E+01	5.17E+01	6.23E+01
	Std. Dev.	2.96E+00	4.09E-01	5.70E-01	4.02E-01	4.54E-01

affects the performance and computation complexity of this optimizer. To study the influences of *NP* on the performance and computation time of MSAPSO, this parameter is set to linearly vary from 40 to 120 with a step size of 20. The numerical simulation results of different metrics for Function F_1 under different settings of NP are reported in Tables 7-8. Fig. 4 displays the obtained Pareto fronts and computation time of different NP settings for Function F_1.

It can be observed from Table 7 that the four performance metrics are averagely better with the increasing of *NP*. This indicates that the obtained Pareto front exhibits an average better performance with a greater value of *NP*. However, one can readily note from Table 8 that the computation time of the optimizer increases significantly with a bigger value of *NP*. It is noticeable from Table 7 that the differences in average values of the three performance metrics between the case where *NP* equals 100 and that of 120 could be negligible. Yet, as shown in Table 8, the computation time consumed by the optimizer is significantly less when $NP = 100$, compared to that of the case where $NP = 120$. Thus, the swarm size is empirically set to 100 for MSAPSO by compromisingly considering the quality of the obtained Pareto front and the computation burden of the optimizer.

B. MAXIMUM ITERATION NUMBER

The maximum iteration number (represented by K_{max}) is a predefined constant in MSAPSO, which can simultaneously affect the quality of the Pareto front and the computation time of the optimizer. In the conducted parameter sensitivity study, K_{max} is linearly changed from 100 to 600 with a step size of 100. Tables 9-10, respectively, summarize the simulation results of the four metrics and computation time with respect to different values of K_{max} for solving Function F_1 using the proposed MSAPSO. The gained Pareto front and computation time under different settings of K_{max} are visualized in Fig. 5.

One can note from Tables 9-10 that the quality of the obtained Pareto front is generally better, whereas the optimizer is more computationally expensive in the case where K_{max} grows bigger. Since the decision maker would prefer to obtain an acceptable Pareto front within a given tackling time in real-world applications, the maximum iteration number is suggested to be 400 in MSAPSO by simultaneously considering trade-offs between the quality of the Pareto front and the computation time.

TABLE 9. Simulation results of different performance metrics for Function F_1 under different settings of the maximum iteration number in MSAPSO.

Items	Results	Maximum iteration number settings					
		100	200	300	400	500	600
NNS	Best	4.90E+01	5.20E+01	4.40E+01	5.10E+01	5.20E+01	5.23E+01
	Average	3.59E+01	3.62E+01	3.66E+01	3.80E+01	3.81E+01	**3.84E+01**
	Worst	2.60E+01	2.50E+01	2.90E+01	2.50E+01	2.52E+01	3.01E+01
	Std. Dev.	6.28E+00	7.42E+00	4.65E+00	6.91E+00	6.33E+00	6.50E+00
GD	Best	3.40E-04	3.05E-04	2.90E-04	2.72E-04	2.66E-04	2.57E-04
	Average	4.57E-04	4.14E-04	3.67E-04	3.61E-04	3.53E-04	**3.45E-04**
	Worst	6.34E-04	4.94E-04	4.88E-04	6.06E-04	4.92E-04	4.78E-04
	Std. Dev.	9.69E-05	6.18E-05	6.99E-05	1.11E-04	7.98E-05	7.22E-05
SM	Best	2.30E-03	2.23E-03	2.03E-03	1.70E-03	2.19E-03	1.96E-03
	Average	3.34E-03	3.22E-03	2.70E-03	2.55E-03	2.56E-03	**2.54E-03**
	Worst	5.25E-03	4.05E-03	4.84E-03	3.40E-03	3.39E-03	3.55E-03
	Std. Dev.	9.00E-04	6.12E-04	8.81E-04	5.15E-04	5.71E-04	5.29E-04

TABLE 10. Simulation results of computation time (*CT* in second) for Function F_1 under different settings of the maximum iteration number in MSAPSO.

Item	Results	Maximum iteration number settings					
		100	200	300	400	500	600
CT	Best	1.72E+01	2.02E+01	2.32E+01	2.54E+01	3.44E+01	3.82E+01
	Average	**1.68E+01**	2.09E+01	2.43E+01	2.61E+01	3.64E+01	3.92E+01
	Worst	1.64E+01	2.20E+01	2.52E+01	2.92E+01	3.72E+01	3.98E+01
	Std. Dev.	3.25E-01	5.03E-01	6.33E-01	3.01E-01	6.10E-01	4.82E-01

TABLE 11. Simulation results of different performance metrics for Function F_1 under different settings of inertia weight in MSAPSO.

Items	Results	Inertia weight				
		0.8	0.7	0.6	0.5	0.4
NNS	Best	4.00E+01	5.10E+01	5.80E+01	5.20E+01	5.70E+01
	Average	3.14E+01	4.32E+01	4.64E+01	4.84E+01	5.08E+01
	Worst	2.70E+01	3.50E+01	3.70E+01	4.50E+01	4.50E+01
	Std. Dev.	5.86E+00	5.67E+00	7.83E+00	2.97E+00	4.76E+00
GD	Best	5.09E-04	6.20E-04	3.97E-04	3.80E-04	3.80E-04
	Average	7.30E-04	5.59E-04	5.38E-04	4.69E-04	4.18E-04
	Worst	1.00E-03	6.70E-04	7.11E-04	5.55E-04	4.40E-04
	Std. Dev.	1.54E-04	6.42E-05	1.41E-04	6.66E-05	2.32E-05
SM	Best	3.83E-03	3.85E-03	2.95E-03	3.08E-03	3.11E-03
	Average	4.50E-03	4.24E-03	4.10E-03	3.78E-03	3.31E-03
	Worst	6.04E-03	5.09E-03	5.32E-03	4.35E-03	3.45E-03
	Std. Dev.	9.29E-04	4.91E-04	1.05E-03	5.16E-04	1.45E-04

TABLE 12. Simulation results of different performance metrics for Function F_1 under different settings of cognitive acceleration parameter in MSAPSO.

Items	Results	Cognitive acceleration parameter									
		2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2
NNS	Best	4.70E+01	4.80E+01	4.50E+01	4.40E+01	4.40E+01	4.40E+01	4.70E+01	5.40E+01	4.90E+01	5.00E+01
	Average	3.68E+01	3.74E+01	3.75E+01	3.78E+01	3.85E+01	3.90E+01	3.92E+01	3.94E+01	4.03E+01	4.04E+01
	Worst	3.00E+01	2.90E+01	3.30E+01	3.20E+01	3.30E+01	3.10E+01	3.40E+01	3.10E+01	3.30E+01	3.10E+01
	Std. Dev.	4.21E+00	5.56E+00	3.98E+00	4.37E+00	4.12E+00	4.47E+00	4.26E+00	6.20E+00	4.76E+00	6.13E+00
GD	Best	3.65E-04	3.20E-04	3.03E-04	2.14E-04	3.18E-04	2.99E-04	3.23E-04	2.38E-04	3.45E-04	2.94E-04
	Average	4.88E-04	4.62E-04	4.60E-04	4.41E-04	4.08E-04	4.03E-04	3.99E-04	3.93E-04	3.89E-04	3.85E-04
	Worst	5.91E-04	6.54E-04	9.68E-04	8.33E-04	5.00E-04	6.30E-04	6.38E-04	6.34E-04	5.90E-04	5.20E-04
	Std. Dev.	7.53E-05	1.18E-04	1.97E-04	1.70E-04	6.79E-05	1.06E-04	1.08E-04	1.19E-04	7.17E-05	7.16E-05
SM	Best	2.79E-03	2.21E-03	2.55E-03	1.79E-03	2.23E-03	2.20E-03	2.65E-03	1.93E-03	2.30E-03	2.25E-03
	Average	3.67E-03	3.64E-03	3.46E-03	3.35E-03	3.09E-03	3.02E-03	3.01E-03	2.99E-03	2.93E-03	2.87E-03
	Worst	4.55E-03	8.71E-03	5.06E-03	4.83E-03	6.32E-03	3.89E-03	4.85E-03	4.76E-03	4.73E-03	4.27E-03
	Std. Dev.	5.49E-04	1.90E-03	9.39E-04	8.45E-04	1.27E-04	5.34E-04	6.55E-04	8.80E-04	7.45E-04	5.82E-04

TABLE 13. Simulation results of different performance metrics for Function F_1 under different settings of social acceleration parameter in MSAPSO.

Items	Results	Social acceleration parameter									
		0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
NNS	Best	1.90E+01	1.80E+01	2.30E+01	2.90E+01	3.30E+01	4.40E+01	4.70E+01	5.00E+01	5.20E+01	5.00E+01
	Average	1.11E+01	1.32E+01	1.54E+01	2.12E+01	2.33E+01	3.31E+01	3.63E+01	3.70E+01	3.73E+01	3.89E+01
	Worst	7.00E+00	9.00E+00	8.00E+00	7.00E+00	1.70E+01	2.70E+01	2.50E+01	2.70E+01	2.70E+01	3.30E+01
	Std. Dev.	3.67E+00	2.57E+00	4.58E+00	6.75E+00	5.54E+00	5.55E+00	6.55E+00	6.98E+00	8.07E+00	5.00E+00
GD	Best	9.24E-04	8.83E-04	8.77E-04	5.27E-04	5.20E-04	3.55E-04	2.52E-04	2.87E-04	2.40E-04	2.33E-04
	Average	1.32E-03	1.20E-03	1.20E-03	8.93E-04	8.18E-04	5.50E-04	4.62E-04	4.13E-04	3.89E-04	3.80E-04
	Worst	1.70E-03	1.54E-03	1.56E-03	2.39E-03	1.08E-03	8.37E-04	7.48E-04	5.63E-04	7.08E-04	5.37E-04
	Std. Dev.	2.49E-04	1.99E-04	2.36E-04	5.42E-04	1.41E-04	1.69E-04	1.60E-04	9.97E-04	1.35E-04	1.01E-04
SM	Best	5.83E-03	6.66E-03	7.02E-03	3.59E-03	4.19E-03	2.50E-03	1.95E-03	2.28E-03	1.95E-03	1.94E-03
	Average	9.35E-03	9.22E-03	8.96E-03	6.66E-03	6.59E-03	4.46E-03	3.60E-03	3.26E-03	3.03E-03	2.94E-03
	Worst	1.10E-02	1.20E-02	1.18E-02	1.90E-02	8.12E-03	7.19E-03	6.36E-03	4.57E-03	6.13E-03	4.12E-03
	Std. Dev.	1.58E-03	2.05E-03	1.53E-03	4.55E-03	1.09E-03	1.55E-03	1.40E-03	8.08E-03	1.23E-03	8.40E-03

C. THREE MAIN CONTROL PARAMETERS

The three main control parameters, namely, the inertia weight (denoted by ω), the cognitive acceleration parameter (represented by c_1), and the social acceleration parameter (denoted by c_2), profoundly affect the exploration and exploitation abilities of particles in MSAPSO. Thus, the quality of the Pareto front heavily relies on these three parameters. It is worth noting in the parameter sensitivity study that: (1) ω decreases from 0.8 to 0.4 with a step size of 0.1; (2) c_1 decreases from 2 to 0.2 with a step size of 0.2 and (3) c_2 increases from 0.2 to 2 with a step size of 0.2. Also, for the case where one parameter is investigated, the remaining two parameters are updated based on the self-adaptive parameter updating rule defined by Eqs. (12)-(15) in Section IV-A.

Tables 11-13 summarize the simulation results of the three performance metrics for Function F_1 under different settings of the three control parameters, respectively. Fig.6 displays the obtained Pareto fronts for Function F_1 under different settings of the three control parameters. We can observe from Tables 11-13 that the average performance of the proposed optimizer heavily relies on values of the three main control parameters of each particle. The proposed MO PSO approach can provide a formidable performance when the inertia weight and cognitive acceleration parameter decrease, whereas the social acceleration parameter increases. Such an observation could be probably interpreted by the facts that: (1) the exploration ability of the proposed method may benefit more from small values of inertia weight and the cognitive acceleration parameter; (2) the exploitation ability of the proposed approach could be more likely promoted for greater values of the social acceleration parameter.

Note that the observation noted above complies with discoveries of some currently-existing works concentrating on improving PSO algorithms by setting different parameter updating strategies [31], [32], [33]. Thus, simultaneously considering the convergence condition given by Eq. (17) and the above simulation results given above, we have empirically set that $\omega_{max} = 0.8$, $\omega_{min} = 0.4$, $c_{1s} = c_{2f} = 2$ and $c_{1f} = c_{2s} = 0.2$ for our proposed PSO optimizer. This parameter setting can not only sufficiently ensure the convergence of our proposed PSO algorithm, but may well balance the exploration and exploitation capabilities of the proposed optimizer.

VIII. CONCLUSION AND FUTURE WORKS

To obtain high-quality Pareto fronts for MO problems, a novel PSO algorithm (called MSAPSO) is first proposed in this study. Attempt to well balance the exploration and exploitation abilities of the proposed algorithm, a self-adaptive parameter updating rule is developed to tune the three key control parameters of each particle. Also, we have investigated the convergence of MSAPSO with respect to different values of the three control parameters due to the fact that the convergence is of great importance in applications of PSO on MO issues. Subsequently, a parameter selection principle is provided to sufficiently ensure the convergence of the proposed PSO.

Utilizing the proposed algorithm, this paper designs an MSPAO-based MO approach, in which a fixed-size external archive is designed to preserve the non-dominated solutions searched by particles. To well distribute the Pareto front, the circular sorting method is combined with the elitist-preserving approach to renovate the external archive. The performance of the proposed method is evaluated by 16 MO test functions against 4 well-known MO methods. The simulation results confirm that the proposed method significantly dominates the four compared MO methods at a confidential level of 95% over the 16 test functions. This indicates that the proposed method can be regarded as an alternative for handling MO problems.

The proposed method and results shown in this paper raise several interesting aspects that deserve some future studies. Firstly, the proposed method could be compared with more state-of-the-art versions of MOEA/D (such as MOEA/D-AAP and MOEA/D-SSC) in order to further verify its effectiveness. Secondly, the proposed could be integrated with more advanced decomposition strategies, such as R2 indicator [19], to promote its convergence and diversity over some complicated MO issues, such as DLTZ problems with more than three objectives. Last but not least, the second-order convergence of the proposed PSO could be investigated in order to provide some insights on improving the proposed method.

APPENDIX A

This Appendix analytically investigates the convergence of MSAPSO. Note that since each dimension in velocity and position vectors of each particle in MSAPSO is independently updated from others, the motion rule as shown by Eqs. (5)-(6) in this algorithm can be simplified into a one-dimensional dynamic system as follows:

$$\begin{bmatrix} X_m(k+1) \\ V_m(k+1) \end{bmatrix} = \begin{bmatrix} 1-c & \omega \\ -c & \omega \end{bmatrix} \begin{bmatrix} X_m(k) \\ V_m(k) \end{bmatrix} + \begin{bmatrix} c \\ c \end{bmatrix} P \tag{22}$$

where:

$$c = c_1 r_1 + c_2 r_2 \tag{23}$$

$$P = \frac{c_1 r_1 \cdot pbest_m^k + c_2 r_2 \cdot gbest}{c_1 r_1 + c_2 r_2} \tag{24}$$

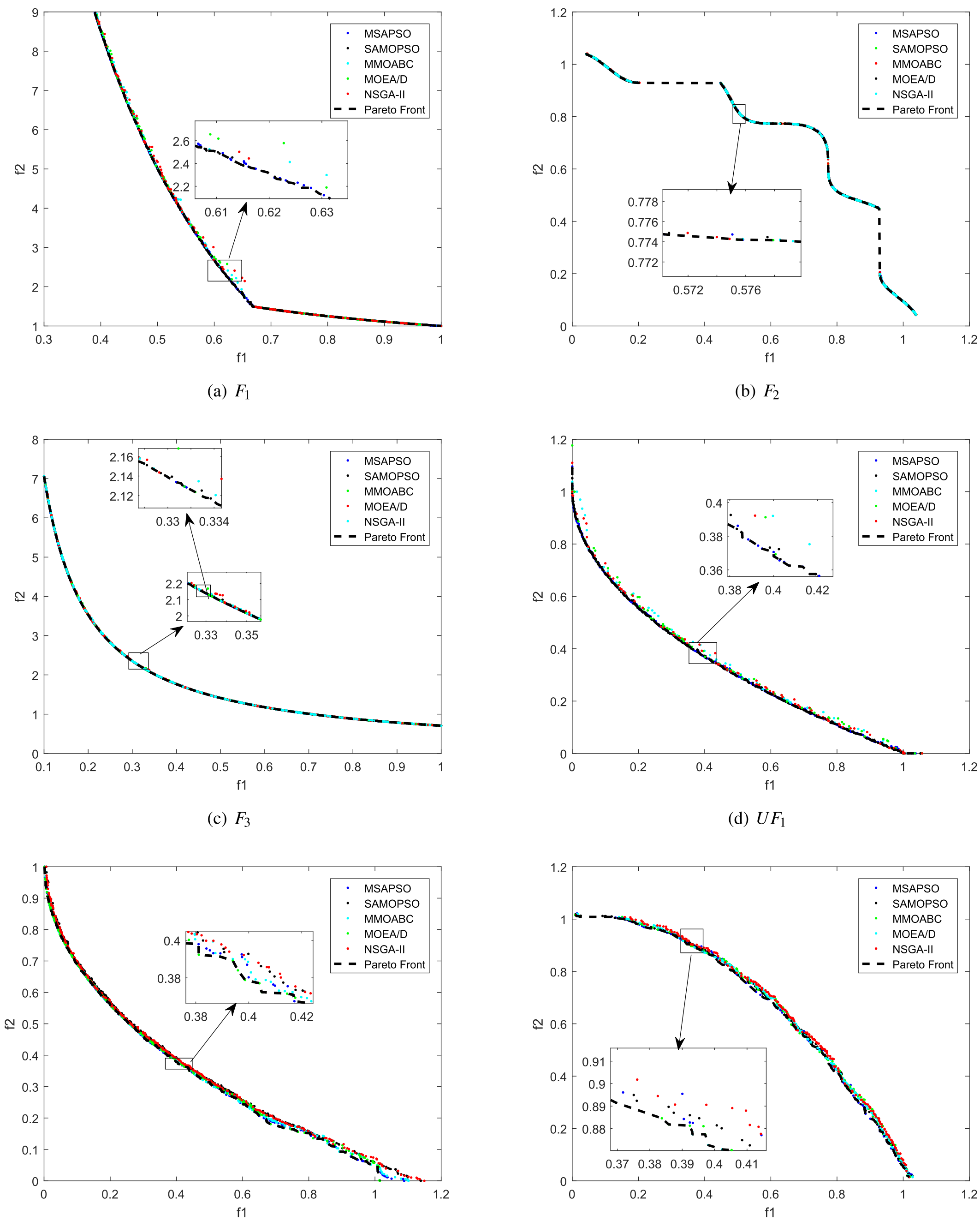
Let $\lambda_{1,2}$ be the characteristic roots to the above dynamic system. Then, we can readily obtain that the characteristic equation and roots to this system as:

$$\lambda^2 - (1+\omega - c)\lambda + \omega = 0 \tag{25}$$

$$\lambda_{1,2} = \frac{1+\omega - c \pm \sqrt{(1+\omega-c)^2 - 4\omega}}{2} \tag{26}$$

Clearly, the dynamic system represented by Eq. (22) converges, if and only if:

$$\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1 \tag{27}$$



(a) F_1 (b) F_2 (c) F_3 (d) UF_1 (e) UF_2 (f) UF_4

FIGURE 7. **Pareto fronts searched by different methods for different test functions.**

Since the characteristic roots $\lambda_{1,2}$ can be real or complex, both these two cases are, respectively, discussed in the following contents.

(a) For the case where $\lambda_{1,2}$ are complex, namely, $\lambda_{1,2} \in \mathbb{C}$, where $\mathbb{C}$ is the imaginary domain.

Lemma 2: For the dynamic system defined by Eq. (22), $\lambda_{1,2} \in \mathbb{C}$, if and only if:

$$\begin{cases} 1+\omega_m - 2\sqrt{\omega_m} < c < 1+\omega_m + 2\sqrt{\omega_m} \\ \omega_m \geq 0 \end{cases} \tag{28}$$

Proof: It is clear from the characteristic equation given by Eq. (25) that $\lambda_{1,2}$ are two complex roots, if and only if:

$$(1+\omega_m - c)^2 - 4\omega_m < 0 \tag{29}$$

Lemma 2 is easily proved by expanding Eq. (29).

Lemma 3: For the case where $\lambda_{1,2} \in \mathbb{C}$, the dynamic system given by Eq. (22) converges, if and only if:

$$\begin{cases} 1+\omega - 2\sqrt{\omega} < c < 1+\omega + 2\sqrt{\omega} \\ 0 \leq \omega < 1 \end{cases} \tag{30}$$

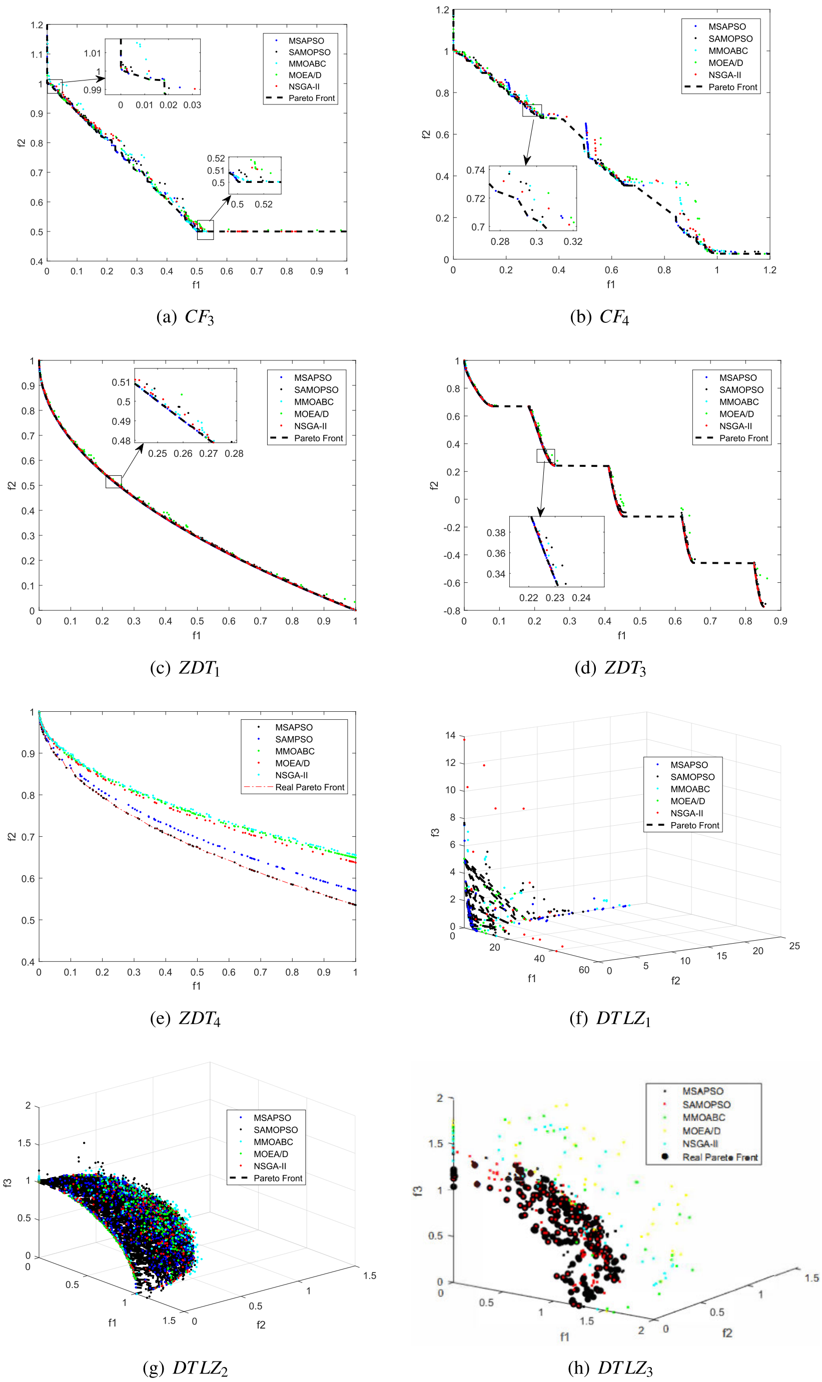


FIGURE 8. **Pareto fronts searched by different methods for different test functions.**

Proof: It is clear from Eq. (26)that, for $\lambda_{1,2} \in \mathbb{C}$, we have:

$$\text{Max}\{|\lambda_1|, |\lambda_2|\} = |\lambda_1| = |\lambda_2| = \sqrt{\omega} \tag{31}$$

Thus, in such a case, $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$ holds, if and only if:

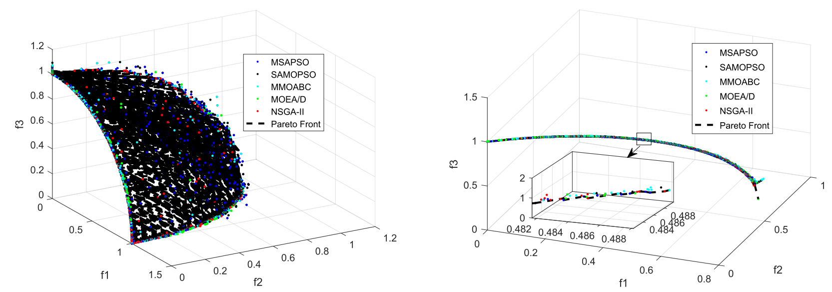
$$\sqrt{\omega} < 1 \tag{32}$$

Considering conditions given by **Lemma 2** and $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$, it can be easily concluded that the dynamic system denoted by Eq. (22) converges for the case where $\lambda_{1,2} \in \mathbb{C}$, if and only if:

$$\begin{cases} 1 + \omega - 2\sqrt{\omega_m} \leq c \leq 1 + \omega + 2\sqrt{\omega} \\ 0 \leq \omega < 1 \end{cases} \tag{33}$$

This completes the proof of **Lemma 3**.

(b) For the case where $\lambda_{1,2}$ are real roots, namely, $\lambda_{1,2} \in \mathbb{R}$, where $\mathbb{R}$ means the real-valued domain.



(a) $DTLZ_4$ (b) $DTLZ_5$

FIGURE 9. Pareto fronts searched by different methods for different test functions.

Lemma 4: For the dynamic system given by Eq. (22), $\lambda_{1,2}$ are two real roots, if and only if:

$$\begin{cases} c \in \mathbb{R},\ \omega < 0 \\ c \leq 1+\omega-2\sqrt{\omega} \text{ or } c \geq 1+\omega+2\sqrt{\omega},\ \omega \geq 0 \end{cases} \tag{34}$$

Proof: It is evident from Eq. (25) that $\lambda_{1,2}$ are two real roots, if and only if:

$$(1+\omega-c)^2-4\omega \geq 0 \tag{35}$$

The proof of **Lemma 4** can be readily completed by expanding Eq. (35).

Lemma 5: For $\lambda_{1,2} \in \mathbb{R}$, the dynamic system given by Eq. (22) converges, if and only if:

$$\begin{cases} 0 < c < 2\omega+2,\ -1 < \omega < 0 \\ 0 < c \leq 1+\omega-2\sqrt{\omega} \text{ or} \\ \ 1+\omega+2\sqrt{\omega} \leq c < 2\omega+2,\ 0 \leq \omega < 1 \end{cases} \tag{36}$$

Proof: For any $\lambda_{1,2} \in \mathbb{R}$, it is trivial from Eqs. (26)-(27) that $\mathrm{Max}\{|\ \lambda_1\ |, |\ \lambda_2\ |\} < 1$ meets, if and only if:

$$-1 < \frac{1+\omega-c \pm \sqrt{(1+\omega-c)^2-4\omega}}{2} < 1 \tag{37}$$

It is clear that, for $\lambda_{1,2} \in \mathbb{R}$, Eq. (37) can be rewritten as follows:

$$c-\omega-3 < \pm\sqrt{(1+\omega-c)^2-4\omega} < c-\omega+1 \tag{38}$$

By expanding Eq. (38), we can have that $\mathrm{Max}\{|\ \lambda_1\ |, |\ \lambda_2\ |\} < 1$ holds in the case where $\lambda_{1,2} \in \mathbb{R}$, if and only if:

$$\begin{cases} 2\omega+2-c > 0 \\ c > 0 \end{cases} \tag{39}$$

Simultaneously considering conditions given by **Lemma 4** and $\mathrm{Max}\{|\ \lambda_1\ |, |\ \lambda_2\ |\} < 1$ given by Eq. (39), it can be easily proven that **Lemma 5** is satisfied in the case where $\lambda_{1,2} \in \mathbb{R}$.

Lemma 6: The dynamic system given by Eq. (22) converges in any value domain, if and only if:

$$\begin{cases} 0 < c_1 r_1 + c_2 r_2 < 2\omega+2 \\ -1 < \omega < 1 \end{cases} \tag{40}$$

Proof: Integrating conditions given by **Lemma 3** and **Lemma 5** together, it is trivial that the dynamic system given by Eq. (22) converges in any value domain, if and only if:

$$\begin{cases} 0 < c < 2\omega+2 \\ -1 < \omega < 1 \end{cases} \tag{41}$$

Substituting Eq. (23) into Eq. (41), the proof of **lemma 6** can be easily completed.

APPENDIX B

This Appendix shows the obtained front of each method for each test function, as illustrated in Figs. 7-9.

REFERENCES

[1] F. Wang, L. Zhou, H. Ren, X. Liu, S. Talari, M. Shafie-Khah, and J. P. S. Catalão, "Multi-objective optimization model of source–load–storage synergetic dispatch for a building energy management system based on TOU price demand response," *IEEE Trans. Ind. Appl.*, vol. 54, no. 2, pp. 1017–1028, Mar. 2018.

[2] H. Hu, X. Sun, B. Zeng, D. Gong, and Y. Zhang, "Enhanced evolutionary multi-objective optimization-based dispatch of coal mine integrated energy system with flexible load," *Appl. Energy*, vol. 307, Feb. 2022, Art. no. 118130.

[3] I. Amallynda and B. Santosa, "Solving multi-objective modified distributed parallel machine and assembly scheduling problem (MDPMASP) with eligibility constraints using metaheuristics," *Prod. Manuf. Res.*, vol. 10, no. 1, pp. 198–225, Dec. 2022.

[4] S. K. Valluru and M. Singh, "Optimization strategy of bio-inspired metaheuristic algorithms tuned PID controller for PMBDC actuated robotic manipulator," *Proc. Comput. Sci.*, vol. 171, pp. 2040–2049, 2020.

[5] Y. Huo, Y. Zhuang, J. Gu, and S. Ni, "Elite-guided multi-objective artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 32, pp. 199–210, Jul. 2015.

[6] Y. Li, M. Xiong, Y. He, J. Xiong, X. Tian, and P. Mativenga, "Multi-objective optimization of laser welding process parameters: The trade-offs between energy consumption and welding quality," *Opt. Laser Technol.*, vol. 149, May 2022, Art. no. 107861.

[7] Y.-F. Huang and S.-H. Chen, "Solving multi-objective optimization problems using self-adaptive harmony search algorithms," *Soft Comput.*, vol. 24, no. 6, pp. 4081–4107, Mar. 2020.

[8] A. Pan, L. Wang, W. Guo, H. Ren, and Q. Wu, "Heuristic orientation adjustment for better exploration in multi-objective optimization," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4757–4771, May 2020.

[9] J. Ning, B. Zhang, T. Liu, and C. Zhang, "An archive-based artificial bee colony optimization algorithm for multi-objective continuous optimization problem," *Neural Comput. Appl.*, vol. 30, no. 9, pp. 2661–2671, Nov. 2018.

[10] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "*R*2-based multi/many-objective particle swarm optimization," *Comput. Intell. Neurosci.*, vol. 2016, Aug. 2016, Art. no. 1898527.

[11] X. Chen, C. Shi, A. Zhou, B. Wu, and Z. Cai, "A decomposition based multiobjective evolutionary algorithm with semi-supervised classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 797–804.

[12] J. Qiao, H. Zhou, C. Yang, and S. Yang, "A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty," *Appl. Soft Comput.*, vol. 74, pp. 190–205, Jan. 2019.

[13] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Inf. Sci.*, vol. 427, pp. 63–76, Feb. 2018.

[14] L. Li, G. Li, and L. Chang, "A many-objective particle swarm optimization with grid dominance ranking and clustering," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106661.

[15] I. C. García, C. A. C. Coello, and A. Arias-Montaño, "MOPSOhv: A new hypervolume-based multi-objective particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 266–273.

[16] F. Li, J. Liu, S. Tan, and X. Yu, "R2-MOPSO: A multi-objective particle swarm optimizer based on R2-indicator and decomposition," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 3148–3155.

[17] L.-X. Wei, X. Li, R. Fan, H. Sun, and Z.-Y. Hu, "A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator," *IEEE Access*, vol. 6, pp. 14710–14721, 2018.

[18] X. Chen, C. Shi, A. Zhou, B. Wu, and P. Sheng, "On balancing neighborhood and global replacement strategies in MOEA/D," *IEEE Access*, vol. 7, pp. 45274–45290, 2019.

[19] F. Li, Z. H. Wu, K. R. Liu, and E. Q. Ge, "R2 indicator and objective space partition based many-objective particle swarm optimizer," *Control Decis.*, vol. 36, no. 9, pp. 2085–2094, 2021.

[20] L. Li, L. Chang, T. Gu, W. Sheng, and W. Wang, "On the norm of dominant difference for many-objective particle swarm optimization," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 2055–2067, Apr. 2021.

[21] J. Liu, F. Li, X. Kong, and P. Huang, "Handling many-objective optimisation problems with R2 indicator and decomposition-based particle swarm optimiser," *Int. J. Syst. Sci.*, vol. 50, no. 2, pp. 320–336, Jan. 2019.

[22] L. Wei, X. Li, and R. Fan, "A new multi-objective particle swarm optimisation algorithm based on R2 indicator selection mechanism," *Int. J. Syst. Sci.*, vol. 50, no. 10, pp. 1920–1932, Jul. 2019.

[23] K. Zou, Y. Liu, S. Wang, N. Li, and Y. Wu, "A multiobjective particle swarm optimization algorithm based on grid technique and multistrategy," *J. Math.*, vol. 2021, pp. 1–17, Dec. 2021.

[24] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[25] B. Tang, Z. Zhu, H.-S. Shin, A. Tsourdos, and J. Luo, "A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm," *Inf. Sci.*, vol. 420, pp. 364–385, Dec. 2017.

[26] L. Wang, Z. Li, C. D. Adenutsi, L. Zhang, F. Lai, and K. Wang, "A novel multi-objective optimization method for well control parameters based on PSO-LSSVR proxy model and NSGA-II algorithm," *J. Petroleum Sci. Eng.*, vol. 196, Jan. 2021, Art. no. 107694.

[27] T. Wang, L. Mo, Q. Shen, Y. B. Zou, and M. Yi, "Multi-objective ecological operation of cascade reservoirs based on MGCL-PSO algorithm," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 612, no. 1, Dec. 2020, Art. no. 012025.

[28] C. Leboucher, H.-S. Shin, P. Siarry, S. L. Ménec, R. Chelouah, and A. Tsourdos, "Convergence proof of an enhanced particle swarm optimisation method integrated with evolutionary game theory," *Inf. Sci.*, vols. 346–347, pp. 389–411, Jun. 2016.

[29] 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.

[30] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, 1995, pp. 39–43.

[31] M. J. Kim, H.-Y. Song, J.-B. Park, J.-H. Roh, S. U. Lee, and S.-Y. Son, "An improved mean-variance optimization for nonconvex economic dispatch problems," *J. Electr. Eng. Technol.*, vol. 8, no. 1, pp. 80–89, Jan. 2013.

[32] R. Akbari and K. Ziarati, "A rank based particle swarm optimization algorithm with dynamic adaptation," *J. Comput. Appl. Math.*, vol. 235, no. 8, pp. 2694–2714, Feb. 2011.

[33] 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, Jun. 2004.

[34] Z. Wang, S. Li, and Z. Sang, "A new constraint handling method based on the modified Alopex-based evolutionary algorithm," *Comput. Ind. Eng.*, vol. 73, pp. 41–50, Jul. 2014.

[35] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, Mar. 2013.

[36] K. Khalili-Damghani, A.-R. Abtahi, and M. Tavana, "A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems," *Rel. Eng. Syst. Saf.*, vol. 111, pp. 58–75, Mar. 2013.

[37] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Inf. Sci.*, vol. 177, no. 22, pp. 5033–5049, Nov. 2007.

[38] J. Zhang, Q. Tang, P. Li, D. Deng, and Y. Chen, "A modified MOEA/D approach to the solution of multi-objective optimal power flow problem," *Appl. Soft Comput.*, vol. 47, pp. 494–514, Oct. 2016.

[39] M. Wang, L. Wang, X. Xu, Y. Qin, and L. Qin, "Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China," *J. Adv. Transp.*, vol. 2019, pp. 1–12, Mar. 2019.

KAIYANG YIN received the M.E. and Ph.D. degrees from the Wuhan University of Technology, Wuhan, China, in 2015 and 2020, respectively. He is currently a Lecturer with the School of Electrical and Mechanical Engineering, Pingdingshan University, Pingdingshan, China. His research interests include optimization algorithm, bipedal robots control, and fuzzy logic systems.

BIWEI TANG received the Ph.D. degree in aircraft design from the School of Astronautics, Northwestern Polytechnical University, China, in 2018. He is currently an Associate Professor with the School of Automation, Wuhan University of Technology, China. His current research interest includes the optimization and control of human-exoskeleton interaction.

MING LI received the bachelor's degree in mechanical engineering and automation from the Hefei University of Technology, Hefei, China, in 2014, the master's degree in mechanical engineering and automation from Anhui Polytechnic University, Wuhu, China, in 2017, and the Ph.D. degree from the School of Automation, Wuhan University of Technology, Wuhan, China, in 2021. He has been with the School of Economics and Management, Anhui Polytechnic University, since 2021. His current research interests include manufacturing systems intelligent optimization and production scheduling.

HUANLI ZHAO received the M.E. degree in testing technology and automation equipment from Northeastern University, Shenyang, China, in 2013. She is currently a Lecturer with the School of Electrical and Mechanical Engineering, Pingdingshan University, Pingdingshan, China. Her research interests include automatic detection technology and optimization and control of robots.