Information Sciences 583 (2022) 266–287

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multi-objective optimization based on an adaptive competitive swarm optimizer

Weimin Huang, Wei Zhang *

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China

ARTICLE INFO

Article history:
Received 11 May 2021
Received in revised form 3 November 2021
Accepted 12 November 2021
Available online 21 November 2021

Keywords:
Multi-objective optimization
Competitive swarm optimization
Improved optimization mechanism
Adaptive strategy

ABSTRACT

Following two decades of sustained studies, metaheuristic algorithms have made considerable achievements in the field of multi-objective optimization problems (MOPs). However, under most existing metaheuristic frameworks, an improved scheme introduced to address specific defects usually leads to additional problems that need to be solved further. Emerging optimization mechanisms should be considered to break the bottleneck, and an adaptive multi-objective competitive swarm optimization (AMOCSO) algorithm, a promising option for solving MOPs, is proposed in this paper. Firstly, the competitive mechanism is modified so that it can perform well on MOPs, and an improved learning scheme is designed for the winners and the losers, which can greatly enhance the optimization efficiency and balance the convergence and the diversity of the proposed algorithm. Then, an external archive and its maintenance schemes are introduced to prevent the population from degenerating and make the algorithm framework more comprehensive. Moreover, a practical adaptive strategy is proposed to fill the blank of parameter research, and no human factors exist in AMOCSO, which means that an amazing promotion can be achieved in generalization. Finally, abundant experimental studies are carried out, and the results of comparative experiments show that the proposed algorithm has significant advantages over several state-of-the-art algorithms.

1. Introduction

Multi-objective optimization problems (MOPs) are the most common problems in real-world engineering, and the related studies of MOPs have made great progress over the past decades [1–3]. The main characteristics of MOPs can be summarized that multiple objectives need to be optimized simultaneously and optimization conflicts exist among these objectives [4–7]. It determines that the solving algorithm of MOPs should possess both the global exploration ability and the local exploitation ability. Since metaheuristic algorithms perfectly meet these demands, they have prevailed in solving MOPs [8,9]. Multi-objective evolutionary algorithms (MOEAs) are the first batch metaheuristic algorithms that have been successfully applied into MOPs [10]. The performance of these classical MOEAs, such as NSGA-II [11], SPEA2 [12], and MOEA/D [13], was once highly evaluated, and these algorithms play a crucial role in the follow-up research of the multi-objective optimization algorithms (MOAs). Other swarm intelligence algorithms, including particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms, have also been developed repidly for solving MOPs stimulated by the success of MOEAs [14,15].

* Corresponding author.
E-mail address: zwei1563@hpu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.ins.2021.11.031

The above-mentioned mainstream algorithms for MOPs have been proposed for more than two decades, and they have been improved and developed continuously [16]. However, these improved schemes still cannot achieve a comprehensive and satisfactory performance. It is a common phenomenon that one aspect of MOAs is improved, while new drawbacks will appear subsequently due to the introduction of additional strategies. The main reason lies in the deficiencies of the basic optimization mechanisms. For instance, a mutation operator is usually added to PSO to prevent it from falling into the local optimum. However, the addition of mutation leads to some additional problems, such as the increased computational complexity and the indeterminate mutation range. Similarly, other problems will also arise when the continuous improvements are proposed for such additional problems. Therefore, an emerging and efficient optimization mechanism is much more appealing rather than constantly repairing on the original foundation. Many novel metaheuristic algorithms have been put forward [17–21], and some of these algorithms get rid of the basic framework of natural biological population behavior and construct novel optimization systems. Tan and Zhu [17] introduced an optimization algorithm inspired by observing fireworks explosion (FA), and Rashedi et al. [18] proposed an algorithm based on the law of gravity and mass interactions (GSA). In GSA algorithm, the searcher agents are a collection of masses, which interact with each other based on Newtonian gravity and the laws of motion. Similarly, a new metaheuristic mechanism called arithmetic optimization algorithm (AOA) was proposed by Abualigah et al. [19], where the distribution behavior of the main arithmetic operators was employed to establish this algorithm. AOA is compatibly implemented to complete the optimization within a wide range of search spaces. These algorithms mentioned above are based on mathematical principles and physical phenomena. In addition, some metaheuristic algorithms based on swarm intelligence have also been widely concerned. A distributed swarm optimizer with adaptive communication was introduced in [20]. In this algorithm, the master optimizer is mainly responsible for communication with the slave optimizers, while each slave optimizer dominates a swarm to traverse the solution space in the iterative process. Particularly, a novel evolutionary mechanism different from PSO, named competitive swarm optimization (CSO), was first proposed by Cheng and Jin [21]. As soon as CSO algorithm was published, scholars were fascinated by its excellent performance and simple implementation process. Compared with other intelligent swarm optimization algorithms, CSO adopts a much more concise mechanism and can achieve a remarkable performance while ensuring the fast convergence.

The main reason for the excellent performance of CSO derives from the competitive mechanism, which can greatly strengthen the global exploration ability of the algorithm. The competitions are performed between the pairwise particles, and the winners and the losers can be distinguished definitely. Then, the winners directly enter the next generation and the losers learn from the corresponding winners and the social information. The competitive mechanism makes all the particles own a chance that can turn into the potential winners and serve as the instruction for other particles. Consequently, CSO algorithm shows an amazing capability in global exploration [22], and the most powerful manifestation is that the large-scale multimodal problems can be solved successfully by CSO without using the mutation operator [23,24]. Although CSO was first proposed to solve single-objective optimization problems (SOPs), it is also suitable for dealing with more complex optimization problems, including MOPs, due to the powerful optimization ability and outstanding efficiency. Moreover, the indispensable mutation component for other MOAs, such as MOEAs and multi-objective particle swarm optimization algorithms (MOPSOs), can be removed from multi-objective competitive optimization algorithm (MOCSO), which not only makes the algorithm more concise but also preserves the evolutionary organization integrity of MOCSO.

In order to further improve the performance and make the framework of MOCSO more comprehensive, an adaptive competitive swarm optimizer, named AMOCSO, is proposed in this paper. The primary motivation of AMOCSO is to inherit the excellent performance of CSO into MOPs. The main contributions of this paper can be summarized as follows.

1. An improved competitive mechanism, composed of Pareto dominance and numerical comparison, is proposed to extend the applicability of CSO, and the relevant components are modified to ensure that the proposed algorithm is suitable for MOPs. Since numerical comparison between non-dominated solutions may be biased due to the difference scales of objectives and Pareto dominance is the basis for evaluating the merits of the solutions, it is resonable to combine them to enhance the competition accuracy. Therefore, in AMOCSO, if the dominance relationship exists between pairwise particles, the non-dominated particle will be directly regarded as the winner. Otherwise, the numerical comparison is further employed to distinguish the winner and the loser.
2. A redesigned learning scheme is proposed to promote the optimization efficiency and accelerate the optimization process. In the canonical CSO, the winners directly enter the next generation to prevent the elite individuals from degenerating. Since the external archive is added for storing the non-dominated solutions in AMOCSO, the updating scheme can also be formulated for the winners. It is a promising scheme that both convergence and diversity of the particles can be further strengthened by learning from the social information. Therefore, the non-dominated solution with the best diversity is selected as the exemplar for the winners. Correspondingly, the losers can also learn from the non-dominated solution with the best convergence in the social item.
3. A general and difficult problem that the performance of the optimization algorithms is usually limited due to the improper threshold settings can be settled perfectly in this paper. The only parameter of CSO, learning factor ϕ, has distinct impacts on the performance and convergence rate. However, the canonical CSO algorithm only provides the range of learning factor, which is stipulated as $\phi \geqslant 0$. After analyzing the optimization process of CSO, a practical and feasible adaptive strategy is proposed, and an algorithm without the influence of human factors is realized, which means that the proposed algorithm has remarkable generalization.

The remainder of this paper is organized as follows. In Section 2, the canonical CSO algorithm is introduced briefly and the literature review is provided. The mechanism and implementation of the proposed AMOCSO algorithm are described detailedly in Section 3. Three benchmark problem suits and several comparison algorithms are employed to carry out the experimental studies in Section 4. The conclusions are summarized in the last Section.

2. Related works

2.1. Canonical CSO algorithm

The main characteristic of canonical CSO algorithm is the competition between pairwise particles, where two particles are randomly selected from the population to produce a winner and a loser. The winners can be distinguished according to

$$winner(P_i, P_j) = \begin{cases} P_i, & f(X_i) < f(X_j) \\ P_j, & f(X_i) > f(X_j) \end{cases} \tag{1}$$

where P_i and P_j represent the i-th particle and the j-th particle, X_i and X_j are the positions of the i-th particle and the j-th particle respectively, and $f(\cdot)$ is the fitness function.

After the competition, the winners diretly enter the next generation and the losers are updated according to the guidance from the winners and the community. Supposing that P_i is the winner and P_j is the loser, the updating of velocity and position can be expressed as

$$\begin{cases} V_{i,d}^{t+1} = 0, & P_i^t = winner \\ V_{j,d}^{t+1} = R_1^t V_{j,d}^t + R_2^t \left(X_{i,d}^t - X_{j,d}^t \right) + \phi R_3^t \left(\overline{X}^t - X_{j,d}^t \right), & P_j^t = loser \end{cases} \tag{2}$$

$$\begin{cases} X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^{t+1} \\ X_{j,d}^{t+1} = X_{j,d}^t + V_{j,d}^{t+1} \end{cases} \tag{3}$$

where $V_{i,d}$ and $X_{i,d}$ represent the d-th dimensional velocity and position of the i-th particle respectively, $d \in [1, n], n$ is the dimension of decision space, $R_1^t, R_2^t, R_3^t \in [0,1]^n$ are three random vectors, $\overline{X}$ donates the mean position of the population, and ϕ is the learning factor that controls the influence of social component.

CSO is mainly composed of the competitive mechanism and the updating scheme, expressed just as (1)–(3). The competitive mechanism makes all particles can turn into the potential winners and serve as the examplar. Furthermore, it is obvious that CSO algorithm is fairly concise, and only one parameter (the learning factor) need to be determined artificially. Therefore, CSO has not only more efficient global optimization capability because of the scheme of learning from winners, but also much stronger generalization than other metaheuristic algorithms (most metaheuristic algorithms have at least two or even more artificial parameters).

2.2. Literature review

Owning to the excellent performance of CSO algorithm, some scholars immediately carried out follow-up studies as soon as CSO was proposed. Therefore, the canonical CSO has been modified to obtain various suitable versions for solving different problems. The competitive mechanism in CSO ranks the particles into two classes: the winners and the losers. For more careful distinguishing, it is natural to design a multi-grade classification and comprehensive optimization strategy based on the competitive mechanism for the particles. Nayak et al. [25] proposed an inherited competitive swarm optimization algorithm (ICSO), where the particle update strategy was designed according to the inheritance mechanism. ICSO algorithm divides particles into grandfather, father, and child, and two third of the population is updated, instead of only half of the population in the canonical CSO. During the updating process, grandfather acts as a leader, and others acquire instruction from him. Occasionally, the child can also absorb information from his father. Similar to ICSO algorithm, Yin and Ming [26] conducted a more detailed distinction of particles. Then, a reverse learning competitive swarm optimization algorithm based on local search (SW-OBLCSO) was proposed, where the canonical competitive mechanism was extended and four particles participated in each competition. The first-ranked particle directly enters the next generation and is called the winner; the second-ranked particle learns in reverse according to the winner; the third-ranked particle competes and learns from the winner; the last-ranked particle obtains the position of the winner and adds a bias set. SW-OBLCSO designs a more detailed particle class division based on the competitive mechanism, and it can increase the convergence speed and maintain the diversity of the population. Although this improved competitive mechanism, involving multiple particles combined with more complex optimization scheme, promotes the performance of CSO, it cannot be avoided that the artificial parameters and the computational complexity of the algorithm are increased greatly. Moreover, no specific and definite guiding recommendations are provided for most of the additional parameters.

A few of sholars focus on the improvemnet of other components while keeping the original competitive mechanism unchanged. A competitive swarm algorithm with covariance matrix self-adaptation (CSOCMSA) was introduced by Li et al. [27]. In CSOCMSA, the covariance matrix adaptation strategy is designed to select the winner, enhance the search ability, and update the covariance matrix of Gaussian model. Some high-quality particles are utilized to estimate the covariance matrix of Gaussian model. Then, the evolutionary direction information is integrated into the Gaussian model, which can endow CSOCMSA with a proper search direction and can improve its search efficiency. Chen and Tang [28] proposed an improved competitive swarm optimization algorithm (ImCSO), where two improved strategies were introduced for further performance enhancement. One is that a ranking paired learning strategy is adopted to enhance the learning efficiency of the losers, and another is that a differential evolution strategy is used to update and improve the winners. Xiong et al. [29] proposed a simplified competitive swarm optimizer (SCSO), in which the optimization process of the canonical CSO is further simplified. SCSO algorithm reduces the number of generated random vectors for the losers, and cancels the item of losers learning from the mean position of the population. Since this simplified mechanism, SCSO has strong real-time performance, which is suitable for some practical problems with high time requirements. However, the over simplified optimization mechanism cannot meet the high performance requirements in dealing with the complex multimodal problems.

Presently, there is no doubt that the relative studies of CSO are still in its preliminary stage, particularly the application of CSO in solving MOPs, named MOCSO. As far as we know, few studies on MOCSO have been published, and the competitive mechanism was firstly adopted for solving MOPs in [30]. Zhang et al. proposed a competitive mechanism based multi-objective particle swarm optimizer (CMOPSO), where the competition occured between particles and non-dominated solutions. CMOPSO integrates the competitive mechanism into MOPSO and adds the polynomial mutation into the framework. However, the influence of PSO can be hardly observed in CMOPSO and the core idea of PSO is replaced by CSO. Although the algorithm removes one of the three items in the updating formulas of the canonical CSO, the reason and motivation are not provided in detail. In addition, CMOPSO is almost no distinction from the canonical CSO in other components, and the modifications according to the characteristics of MOPs are not considered, which could lose the integrity of optimization process.

Other representative studies have emerged in recent. A large-scale multi-objective optimization based on a competitive swarm optimizer (LMOCSO) was introduced by Tian et al. [31]. A special learning strategy is proposed to enhance the optimization ability, which can make the particles directly move toward the leader and achieve the purpose of fast convergence. Although LMOCSO employs the competitive mechanism, it seems that the competitive mechanism is not the focus of this study. Except for the learning strategy, the rest components of LMOCSO are designed according to the canonical CSO, without much modifications. However, due to the inherent fast convergence rate of CSO, the learning strategy of LMOCSO leads to too fast convergence, which may not be an outstanding scheme in solving MOPs. The main reason lies in that the direct learning strategy greatly weakens the local exploitation ability of particles, which is a fatal impairment for the diversity of LMOCSO [32]. The complicated competitive mechanism involving multiple particles is also applied into MOCSO. Deng et al. [33] proposed a competitive particle swarm algorithm based on vector angles (VaCSO). In this algorithm, three-particle competition mechanism is adopted, and the population is divided into two subgroups, which are endowed with different generation strategies. The convergence and diversity of the algorithm can be improved simultaneously. Moreover, auxiliary learning based on vector angle information is also considered to optimize the performance gap of particles in the population. However, this complicated competitive mechanism that multiple particles participate in each competition could lead to the reduction of leaders in the population and could weaken the global exploration ability of the algorithm.

Different from the above improvements, a hybrid many-objective competitive swarm optimization algorithm (HMaCSO) was proposed by Xue et al. [34]. Two diferent measurements, including adaptive grid division strategy and mating pool selection method, are employed to formulate the mating operation in HMaCSO, named particle grouping. Meanwhile, the self-controlling dominance area of solution strategy (S-CDAS) plays an important role in retaining the non-dominated solutions. Although the environment selection scheme proposed by HMaCSO provides a guarantee for the elite individuals, other improvement schemes, such as particle grouping method, are not of great significance. Generally, the random selection can meet the requirements adequately for the random combination of particles in CSO. Therefore, using other more complex schemes, such as particle grouping, cannot considerably improve the performance of CSO but increase the computational complexity of the algorithm. Compared with the improvement scheme mentioned above, it is more useful to consider from the perspective of particle updating.

Although the research of CSO algorithm has made some progress, there are still some crucial problems that have not been solved. First of all, the competitive mechanism are not explained in the background of MOPs, where all the improved schemes just transplant the competitive mechanism from SOPs to MOPs without any change. Actually, it may not be a wise choice for stimulating the potentiality of competitive mechanism. The competition for two non-dominated individuals is not introduced, and the implementation of the comparison mechanism is different from the standard operating procedures and should be treated specially in MOPs. Secondly, the position of winners will be kept in canonical CSO algorithm to prevent the elite individuals from degenerating. Obviously, it is difficult that just relying on half of the particles strives to accomplish the optimization process in complex MOPs. The external archive can alleviate the degeneration of elite individuals, in order that the update schemes can also be formulated for the winners to accelerate the optimization efficiency of MOCSO. Finally, the learning factor ϕ is the only one parameter, and it is one of the decisive factors affecting the performance of MOCSO. However, the research on the value guidance of ϕ has not been mentioned in existing works. Up to now, only $\phi \geqslant 0$ is provided as the appropriate range for achieving the convergence, and there is no further explanation and guidance about the learning factor. Actually, extensive studies on the parameters have been done in other metaheuristic algorithms, and some effective guiding suggestions have been introduced to stimulate their full performance [35–37]. Therefore, it is also an essential topic to study the parameter settings of CSO.

In response to the above problems, an adaptive multi-objective competitive swarm optimization (AMOCSO) algorithm is proposed in this paper.

3. Proposed AMOCSO

The overall framework of the proposed AMOCSO algorithm and the relationship between its components are presented in Fig. 1. AMOCSO algorithm consists of a backbone and a branch, where the backbone constitutes by three components: the competitive mechanism, the updating schemes, and the external archive. After the optimization has completed, the non-dominated solution set stored in the external archive is taken as the output, which is the outcome of the proposed algorithm. The branch corresponds to the proposed adaptive strategy composed of evolutionary environment detection and parameter adaptor.

The implementation of AMOCSO can be summarized as follows. A pair of particles are randomly selected for the competition to distinguish the winner and the loser, and two updating schemes are provided for them respectively. The obtained non-dominated solutions are stored in the external archive, and a composite index is introduced to maintain the external archive when its capacity reaches the maximum. Meanwhile, the social information, extracted from the competition of particles, is employed as feedback to detect the evolutionary environment and adjust the optimization process. The details of each component are expounded in the following sections.

3.1. Competitive mechanism and updating scheme

The basic competitive mechanism of the canonical CSO algorithm just contains the numerical comparison of two particles, which is simple and for SOPs only. First and foremost, the competitive mechanism should be supplemented to suit the environment of MOPs. Thus, the fundamental principle in solving MOPs, Pareto dominance, is added to the competitive mechanism, and the procedure of particle comparison is presented in Fig. 2.

The two-stage competition is designed to identify the winners and the losers. If the dominated relationship exists between the pairwise particles, the non-dominated particle is the winner and the dominated particle is the loser. Otherwise, the numerical comparison is employed to distinguish the winner from the loser.

Supposing that the particle P_a strictly dominates the particle P_b, denoted as $P_a \prec P_b$. The specific discriminant of Pareto dominance can be presented as

$$\forall i \in [1, M] : f_i(X_a) \leqslant f_i(X_b) \;\; \wedge \;\; \exists j \in [1, M] : f_j(X_a) < f_j(X_b) \tag{4}$$

where $f_i(\cdot)$ is the i-th objective function, X_a and X_b are the positions of P_a and P_b, and M is the number of objectives.

Numerical comparison is the comparison of fitness accumulation on all objectives, which can simply reflect the advantage and drawback of particles on convergence. The numerical value of the i-th particle is

$$F(X_i^t) = \sum_{m=1}^{M} f_m(X_i^t) \tag{5}$$

To facilitate the understanding of the proposed algorithm, the implementation details of the competitive mechanism are presented by pseudo-code, as shown in Table 1.

After the competitions, the position and the velocity of the winners and the losers can be updated respectively to produce the next generation. The merit of canonical competitive mechanism, the losers learn from the winners, is reserved in this

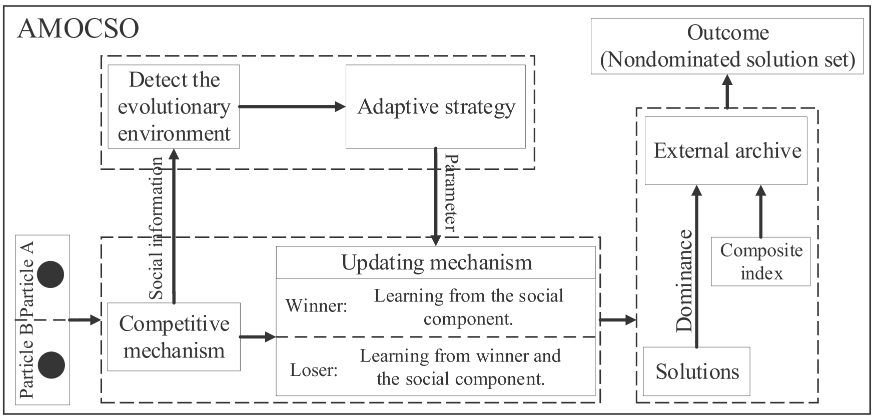

Fig. 1. Framework of AMOCSO algorithm.

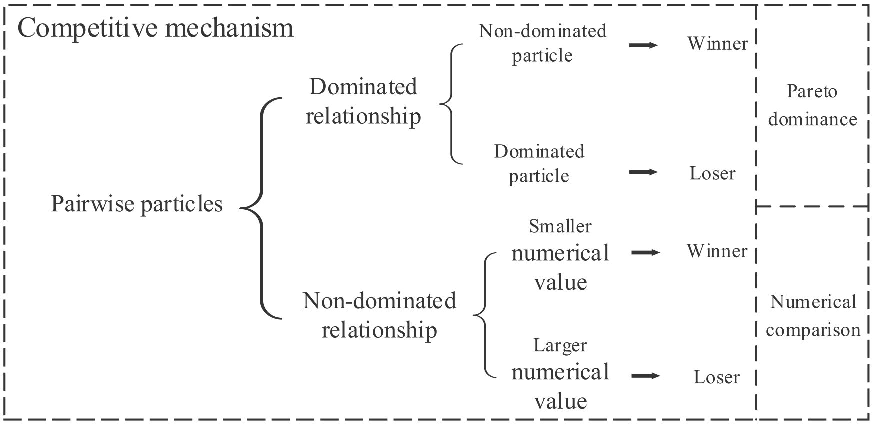

Fig. 2. Improved competitive mechanism for MOPs.

paper. Moreover, the improved social component, learning from the representative non-dominated solutions, is introduced to enhance the convergence of the losers. The updating of losers can be expressed as

$$V^{t+1}_{lose,d} = R^t_1 V^t_{lose,d} + R^t_2 \left(X^t_{win,d} - X^t_{lose,d} \right) + \phi R^t_3 \left(c - NS^t_d - X^t_{lose,d} \right) \tag{6}$$

$$X^{t+1}_{lose,d} = X^t_{lose,d} + V^{t+1}_{lose,d} \tag{7}$$

where V_{win} and V_{lose} represent the velocities of the winner and the loser respectively, X_{win} and X_{lose} denote the positions of the winner and the loser respectively, and $c - NS$ is the non-dominated solution with the best convergence in the external archive.

Correspondingly, the updating scheme is also designed for the winners.

$$V^{t+1}_{win,d} = R^t_4 V^t_{win,d} + \phi R^t_5 \left(d - NS^t_d - X^t_{win,d} \right) \tag{8}$$

$$X^{t+1}_{win,d} = X^t_{win,d} + V^{t+1}_{win,d} \tag{9}$$

where R^t_4 and R^t_5 are also the random vectors similar to R^t_1, R^t_2, and R^t_3, and $d - NS$ is the non-dominated solution with the best diversity in the external archive.

Remark 1. In this paper, the canonical updating scheme of CSO is modified, and the representative non-dominated solutions are introduced to facilitate the social information exchange. The main considerations for designing the learning strategies are as follows. The losers generally perform poorly in convergence that should be improved primarily. Conversely, the winners have a preferable foundation of convergence and the expectation for improving the diversity of the proposed algorithm is naturally undertaken by them. The convergence and diversity of AMOCSO can be well balanced by conducting two directions of learning guidance simultaneously.

3.2. Non-dominated evaluation index

The representative non-dominated solutions with the best performance ($c - NS$ and $d - NS$) are introduced in (6) and (8) to strengthen the convergence and the diversity of AMOCSO. They are selected from the external archive according to the corresponding evaluation indexes, including the convergence index and the diversity index.

In terms of convergence, two evaluation indexes, the average ranking (AR) and the global damage (GD) [38], are used to avoid the problem that incomplete evaluation leads to the particles aggregation. The calculations of AR and GD can be expressed by

$$AR^t(NS_i) = \frac{1}{M} \sum_{m=1}^{M} \frac{Rank^t_m(NS_i)}{N^t} \tag{10}$$

$$GD^t(NS_i) = \frac{1}{N^t M} \sum_{j=1, NS_i \neq NS_j}^{N^t} \sum_{m=1}^{M} \frac{\max\left(f^t_m(NS_i) - f^t_m(NS_j), 0 \right)}{f^t_{m,\max} - f^t_{m,\min}} \tag{11}$$

Table 1
Implementation of the competitive mechanism.

```
A pair of particles are randomly selected and recorded as the i-th particle and the j-th particle.
if  the dominated relationship exists between the i-th particle and the i-th particle
    The non-dominated particle is the winner and the dominated particle is the loser.
else
    Calculate the sum of objectives value.
    if  F(X_i) < F(X_j)
        The i-th particle is the winner and the j-th particle is the loser.
    else
        The j-th particle is the winner and the i-th particle is the loser.
    end
end
```

where NS_i is the i-th non-dominated solution in the external archive, N is the number of non-dominated solutions in the external archive, $Rank_m$ is the rank function of the m-th objective, and $f_{m,max}$ and $f_{m,min}$ are the maximum and minimum on the m-th objective respectively.

The integrated convergence evaluation index is simply constituted by the sum of AR and GD.

$$IC^t(NS_i) = AR^t(NS_i) + GD^t(NS_i) \tag{12}$$

where IC is the abbreviation of integration index. Normalization is adopted in (10) and (11), which ensures the additivity of AR and GD. Then, the non-dominated solutions with the best convergence can be selected by

$$c - NS^t = \min\left(IC^t(NS)\right) \tag{13}$$

Crowding distance [11] is usually employed as the criteria for evaluating the distribution of non-dominated solutions. Taking a solution NS_i in the external archive as an example and supposing that its nearest neighbor solutions on both sides are NS_{i-1} and NS_{i+1}, the crowding distance can be calculated by

$$CD^t(NS_i) = \sum_{m=1}^{M} dist(f_m(NS_{i-1}), f_m(NS_{i+1})) \tag{14}$$

where $dist(f_m(NS_{i-1}), f_m(NS_{i+1}))$ refers to the Euclidean distance between $f_m(NS_{i-1})$ and $f_m(NS_{i+1})$. Then, the non-dominated solutions with the best diversity can be determined by (15).

$$d - NS^t = \max\left(CD^t(NS)\right) \tag{15}$$

In addition, the two evaluation indexes (IC and CD) can also be used to maintain the external archive. A composite evaluation index is designed to assess the comprehensive performance of the non-dominated solutions, which can improve the quality of the solutions which are remained in the external archive. Furthermore, the composite index can avoid some problems caused by using the single index to maintain the external archive. One of the most prevalent schemes, using only the diversity index, may delete the elite solutions accidentally and lead to the degradation of the population. The proposed composite index can be expressed by

$$CI^t(NS_i) = \frac{IC^t(NS_i)}{CD^t(NS_i)} \tag{16}$$

The non-dominated solution with the maximum CI value will be kicked out from the external archive when the external archive reaches the maximum capacity. The detailed implementation of the leader selection and the external archive maintenance is shown in Table 2 with pseudo-code.

Remark 2. The convergence index and the diversity index are introduced to select the social information leaders. The integrated convergence index is composed of two evaluation indexes to promote the comprehensiveness of evaluation, which avoids the aggregation of particles at the edge or the center of Pareto front caused by incomplete evaluation of single convergence index. Moreover, a composite evaluation index, the combination of the convergence index and the diversity index, is employed to maintain the external archive, and it plays an important role in balancing convergence and diversity. The preservation of non-dominated solutions with the better comprehensive performance can be guaranteed and the problems can be well avoided that only depend on one evaluation index.

3.3. Adaptive strategy for parameter adjustment

Similar to most metaheuristic algorithms, the parameter of CSO can seriously affect its performance. In order to minimize the human factors and improve the applicability of AMOCSO, a simple and practical adaptive strategy is proposed for parameter adjustment.

Almost all adaptive parameter adjustment strategies are formulated according to the detection of evolutionary environment [39], and the core idea in detecting the evolutionary environment is to construct some appropriate intermediate variables, such as the entropy in pccsAMOPSO [40] and the spacing information in AMOPSO [16]. The related intermediate variables that can reflect the performance gap between the winners and the losers are established and employed for forecasting the optimization trend of the proposed algorithm.

After the competitions, the winners and the losers of each pairwise particles have been distinguished, and whether there is a dominant relationship between the winners and the losers has also been determined. Based on this premise, the quantization difference of each pairwise particles can be defined as

$$\rho^t(X^t_{win}, X^t_{lose}) = \begin{cases} \frac{1}{M}\sum\limits_{m=1}^{M} \frac{|f_m(X^t_{win}) - f_m(X^t_{lose})|}{f^t_{m,\max} - f^t_{m,\min}}, & X^t_{win} \prec X^t_{lose} \\ 0, & Otherwise \end{cases} \tag{17}$$

Then, the average quantization difference of the population can be obtained by

$$AQ^t = \frac{1}{K}\sum_{k=1}^{K} \rho^t_k\left(X^t_{k,win}, X^t_{k,lose}\right) \tag{18}$$

where $k = 1, 2, \ldots, K, K$ is the number of particles in pairs, i.e., half the number of particles in the population. Obviously, the value of AQ is related to the optimization status of the proposed algorithm. The larger AQ means that the algorithm is in the global exploration, while the smaller AQ indicates that the local exploitation is in dominant. The value of AQ will eventually stabilize and tend to zero, which means that the algorithm has converged steadily. Therefore, the increment of AQ, given in (19), can be used to reflect the evolutionary environment of the proposed algorithm in real time.

$$\Delta AQ^t = AQ^t - AQ^{t-1} \tag{19}$$

In order to meet the convergence conditions and most applicable situations of the proposed algorithm, ϕ is confined to $[0, 2]$. A simple incremental adjustment scheme is applied to the parameter adjustment.

$$\phi^t = \begin{cases} \phi_0 + \frac{|\Delta AQ^{t-1} + \Delta AQ^t|}{2}, & \Delta AQ^t > 0 \\ \phi_0, & \Delta AQ^t = 0 \\ \phi_0 - \frac{|\Delta AQ^{t-1} + \Delta AQ^t|}{2}, & \Delta AQ^t < 0 \end{cases} \tag{20}$$

where, $\phi_0$is the base value of the learning factor and it is usually set to the middle of its upper and lower bounds. The first order inertial filtering is adopted in (20) to enhance the robustness of parameter variation.

To clearly show the detection of evolutionary environment and the adaptive adjustment process of parameter, the experimental results of the proposed AMOCSO algorithm on DTLZ2 (a benchmark problem with tri-objective) are presented in Fig. 3.

Since the parameter ϕ can control the proportion of social information guidance, the larger value of ϕ can strengthen the global exploration and the smaller value of ϕ is beneficial for the local exploitation. Based on the above principles, the parameter adjustment scheme can be understood easily. It can be observed from the experimental results that there are several peaks on the curve of AQ. The rising phases of the peaks indicate that AMOCSO enters the global exploration and the parameter ϕ should be increased correspondingly. Conversely, the descending phases of the peaks are related to the local exploitation and the parameter ϕ should be decreased. AQ can be stable at zero eventually, which indicates that the optimization of the proposed algorithm has been completed.

The implementation details of the proposed adaptive strategy are presented by pseudo-code in Table 3.

Table 2
Implementation of the leader selection and the external archive maintenance.

```
Calculate the convergence index IC(NS).
Calculate the diversity index CD(NS).
Select c-NS from the external archive: c-NS = min(IC(NS)).
Select d-NS from the external archive: d-NS = max(CD(NS)).
Update the velocity and position of particles.
Save the discovered non-dominated solutions to the external archive.
if the number of non-dominated solutions exceeds the capacity of the external archive
    while the number of non-dominated solutions is equal to the maximum capacity of the external archive.
        Calculate the diversity index CI(NS).
        Delete a non-dominated solution: NS_i = max(CI(NS)).
    end
else
    Continue.
end
```

Remark 3. Adaptive strategy has always been considered as an important and interesting topic in metaheuristic algorithms, and an effective adaptive strategy can greatly improve the balance between convergence and diversity of MOAs [41,42]. A feasible and practical adaptive strategy, based on the competitive mechanism, is proposed in this paper, which fills the gaps in parameter research of MOCSO algorithm. The average quantization difference (*AQ*) between the winners and the losers is employed as the feedback information to detect the evolutionary environment, and then the parameter adjustment can be formulated. Moreover, the proposed adaptive strategy does not introduce new parameters, which greatly enhances the generalization of the proposed algorithm.

4. Experimental studies

To test and verify the performance of the proposed AMOCSO algorithm, the most widely used benchmark suites, DTLZ [43], WFG [44], and CEC2009 (UF) [45], are employed as the test problems, in which include 11 bi-objective problems (WFG1-WFG4 and UF1-UF7) and 15 tri-objective problems (WFG5-WFG9, DTLZ1-DTLZ7, and UF8-UF10). Furthermore, seven classical and state-of-the-art MOAs, including CMOPSO [21], cdMOPSO [46], pccsAMOPSO [40], NSGA-II [11], SPEA2 [12], MOEA/D [13], and AUDHEIA [47], are selected as the comparison algorithms. The basic experimental settings of all algorithms are set to the identical conditions in order to ensure the fairness of experiments. The parameter settings of each algorithm are presented in Table 4, which are consistent with those provided in the original literature.

In Table 4, S is the population size, N is the maximum number of non-dominated solutions, η_m is the distribution index of polynomial mutation, and $p_m = 1/n$ is the mutation probability, where n is the number of decision variables. In MOPSO, the inertia weight ω and the learning factors c_1 and c_2, are the basic parameters. γ is the elite population size of CMOPSO. For pccsAMOPSO, $\omega(0), c_1(0)$, and $c_2(0)$ are the initial values of flight parameters and $Step_{\omega}, Step_{c_1}$, and $Step_{c_2}$ are the range intervals of ω, c_1 and c_2 respectively. In MOEA/D, T determines the size of the neighborhood, η_r is the maximum number of child solutions inherit from the parent solutions, and δ defines the chosen probability of parent solutions. For AHUDEIA, CR and F are two control parameters in evolutionary strategy, NA is the size of selected antibodies for cloning proliferation, p_c is the crossover probability, and η_c is the distribution index of SBX. The maximum number of iterations is set to 300. In order to reduce the occasionality, all test instances are run 30 times independently and the relevant statistical results are recorded. All experiments perform on Windows 10 system, CPU frequency is 2.7 GHz, RAM of the computer is 8 GB, and the experimental platform is MATLAB2016b.

4.1. Performance metrics

Two widely used performance metrics, inverted generational distance (IGD) and hypervolume (HV), are employed to quantify the convergence and the diversity of the proposed algorithm.

IGD is generally used to measure the distance between the true Pareto front and the non-dominated solution set obtained by the algorithms. The smaller IGD result indicates that the approximate non-dominated solution set is closer to the true Pareto front, which means the better performance. The specific calculation of IGD can be described as:

$$IGD(PF, PF^*) = \frac{1}{N}\sum_{i=1}^{N} \min(dist(PF_i, PF^*)) \tag{21}$$

where PF is the approximate Pareto front obtained by the algorithm, PF^* is a set of sampling points from the true Pareto front, and N is the number of non-dominated solutions.

HV strictly follows the Pareto dominance principle and has perfect theoretical support [48], so it can be utilized to evaluate the convergence, uniformity, and universality of the non-dominated solution set simultaneously. The larger HV result means that the approximate non-dominated solution set possesses the better comprehensive performance. The format of HV is defined as:

$$HV(Ref, PF) = Leb\left(\bigcup_{NS\in PF} [NS_1, r_1] \times [NS_2, r_2] \times \cdots \times [NS_M, r_M]\right) \tag{22}$$

where $Leb(\cdot)$ is the Lebesgue Measure and $Ref = (r_1, r_2, \ldots, r_M)$ is the reference point in the objective space that is dominated by all the Pareto optimal solutions. The HV metric of PF is defined as the volume of the objective space dominated by solutions in PF and bounded by Ref [49].

4.2. Results discussion

4.2.1. Comparisons on IGD metric

The comparison algorithms and the proposed algorithm can be divided into three categories, which are MOCSOs, MOPSOs, and MOEAs, respectively. The experimental results of IGD metric on DTLZ, WFG, and UF benchmark suits are listed in Tables 5 and 6, including the mean and the standard deviation (Std.). In each test problem, the best experimental results are shown in bold.

It can be clearly observed from Tables 5 and 6 that the proposed AMOCSO algorithm obtains 18 optimal results in all 26 instances, which is an excellent result for the existing MOAs under such comparative conditions. DTLZ test suit, as the most commonly used benchmark problem for testing the performance of MOAs, can pose some difficulty for most MOAs on some problems, such as DTLZ1 and DTLZ3 problems. AMOCSO performs remarkably on DTLZ text suit, outperforms all the comparison algorithms on 5 DTLZ problems, and is only slightly inferior to AUDHEIA on 2 problems (DTLZ5 and DTLZ6 problems). Since WFG problems encompass a diverse set of properties that can be found in real-world MOPs [44], these problems are more complex compared with DTLZ test suit and can raise substantial obstacles for any multi-objective optimization algo-

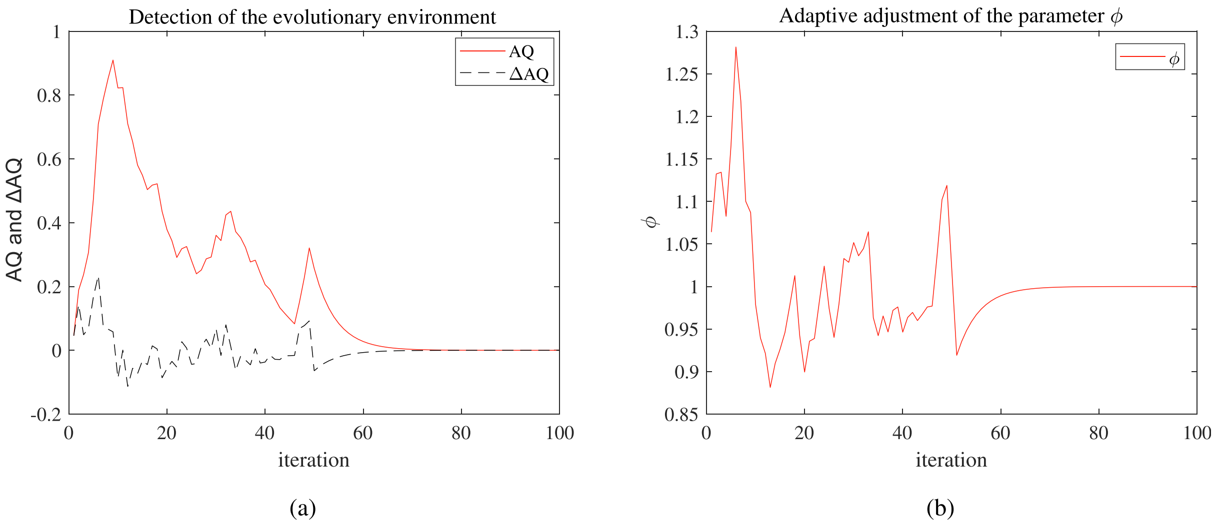

Fig. 3. Adaptive process of the proposed algorithm in solving DTLZ2. (a) Evolutionary environment detected by the proposed adaptive strategy. (b) Adaptive adjustment of the parameter ϕ.

Table 3
Implementation of the adaptive strategy.

$for \quad i = 1 : K$

  Calculate the quantization difference between i-th pairwise particles by (17).

end

Calculate the average quantization difference AQ by (18).

Calculate the incremental of AQ by (19).

Update the parameter according to (20).

rithm. However, AMOCSO equally achieves satisfactory performance on WFG benchmark suit. 7 best experiemntal results are obtained by AMOCSO in all 9 problems and AUDHEIA and CMOPSO obtain the best results of the remaining 2 instances respectively. The experimental results of UF problems are listed in Table 6. UF is a much more complex MOPs family, and it is even difficult for many MOAs to approach the Pareto front in solving some UF problems. Although AMOCSO on UF problems is not as far ahead as on DTLZ and WFG problems, it is still in an absolutely dominant place. The proposed AMOCSO performs best on 6 problems (UF2, UF3, UF5, UF7, UF9, and UF10 problems). The remaining 4 best performance are obtained by MOEA/D on UF1 problem, pccsAMOPSO on UF6 problem, and AUDHEIA on UF4 and UF8 problems, respectively.

Generally, supposing that the mean IGD result of DTLZ and WFG problems reaches 10^{-3}, the non-dominated solution set obtained by the algorithms can be considered as an excellent approximation to the true Pareto front. As can be seen from the overall results of DTLZ and WFG problems that three algorithm, AMOCSO, AUDHEIA, and pccsAMOPSO, obtain 10^{-3} results on most problems. Only on DTLZ3 and WFG5 problems, none of the algorithms achieve the standard of 10^{-3}, and these prolems are also the only two problems that AMOCSO fails to meet such standard. Due to the characteristics of UF test suit, it is difficult to obtain the same excellent results as on DTLZ and WFG problems. However, the proposed AMOCSO algorithm still can achieve the excellent results of 10^{-2} on most of UF problems. Among the selected 7 comparison algorithms, AUDHEIA shows considerable competitiveness on IGD results. The main reason lies in that the state-of-the-art techniques have been carried out in AUDHEIA, which can be summarized as follows. A uniform distribution selection mechanism is proposed in AUDHEIA to enhance its convergence and diversity, and individuals can be mapped to a hyperplane, which is correlated with the objective space and are clustered to increase the performance of the algorithm. Moreover, the mapped hyperplane is sectioned uniformly and a threshold as the judgment criterion is adjusted adaptively.

In order to further evaluate the experimental results between the proposed algorithm and the comparison algorithms, the non-parametric statistical hypothesis testing, Wilcoxon's rank-sum test, is introduced to cope with the experimental results. The statistical significance for IGD metric results of the selected algorithms can be observed intuitively. The symbols "+", "$-$", and "$\approx$" demonstrate that the results of comparison algorithms are superior, inferior, and similar to those obtained by AMOCSO, respectively. The index "Rank" records the ranking of the algorithms on each problem. Statistical results, including "Rank Sum", "Final Rank", and "$+/-/\approx$" indexes, are presented at the end of the tables. It can be easily observed from the results of WFG6 problem in Table 5 that AMOCSO is the second-ranked algorithm and AUDHEIA obtains the best result, while Wilcoxon's rank-sum test reults show that the results of these two algorithms are similar, that is, it can be considered that the two algorithms perform equally excellent on WFG6 problem. For UF problems, Wilcoxon's rank-sum test results show that AMOCSO achieves the similar results compared to CMOPSO on 3 problems, to pccsAMOPSO on 2 problems, to MOEA/D on 2 problems, and to AUDHEIA on 5 problems. The best performed comparison algorithm is AUDHEIA and its entire ranking on UF instances is 24, which is not much difference from the ranking obtained by AMOCSO (the overall ranking of AMOCSO on UF problems is 20). However, hypothesis testing results show that AUDHEIA is inferior to AMOCSO on 4 problems (UF2, UF3, UF5, and UF10 problems). Other competitors have at least 40 entire ranking sum and 7 results are worse than AMOCSO on UF problems. On UF2 problem, the performance of AMOCSO is over 40% better than that of the best performed comparison algorithm (AUDHEIA). Contrarily, the performance of AMOCSO is only 20% behind the first-ranked algorithm (pccsAMOPSO) on the worst performed problem (UF6 problem). AMOCSO in solving three problems, UF2, UF3, and UF5, are completely ahead of the competitors and also has obvious advantages on the other instances. From the overall results, it can be obtained that AMOCSO, AUDHEIA, and pccsAMOPSO take the top three places in all eight algorithms respectively. It is worth mentioning that although the final ranking of pccsAMOPSO is slightly higher than CMOPSO, Wilcoxon's rank-sum test shows that the performance of CMOPSO (the sum of $+/-/\approx$ index obtained by CMOPSO is 1/19/6) is better than pccsAMOPSO (the sum of $+/-/\approx$ index obtained by pccsAMOPSO is 1/20/5).

Based on the above discussion of IGD metric results, we can summarize that the proposed AMOCSO algorithm shows overwhelming superiority on the majority of the test instances, which can demonstrate forcefully that AMOCSO has more remarkable performance compared with these selected comparison algorithms.

In order to intuitively demonstrate the performance of each algorithm, the non-dominated solution set obtained by the selected algorithms is provided in Figs. 4–13. The plotted figures are the nearest set obtained by the selected algorithm after 30 independent runs. The results of bi-objective WFG problems, WFG1-WFG4, are presented in Figs. 4–7. It can be seen conspicuously that the disparity exists between AMOCSO and the competitors. Particularly, the drawbacks of the selected algorithms can be observed clearly on the results of WFG1 and WFG2 problems, where these problems are relatively difficult to converge to the Pareto optimal solution set. For some comparison algorithms, such as cdMOPSO, NSGA-II, SPEA2, and MOEA/D, the obtained non-dominated solution set cannot approach and cover the true Pareto front completely. There is a significant performance gap between these algorithms and AMOCSO. The IGD metric results of AMOCSO are more than 90% ahead of them on WFG1 and WFG2 problems respectively. Although the remaining three comparison algorithms, CMOPSO, pccsAMOPSO, and AUDHEIA, roughly perform similar convergence to AMOCSO, there are still considerable performance gaps in the specific experimental data. On WFG1 problem, the mean IGD result of AMOCSO is over 34% better than that of the best performed comparison algorithm (AUDHEIA). Moreover, AMOCSO has absolute advantages on WFG2 problem and IGD performance gap between AMOCSO and the best performed competitor (CMOPSO) is over 90%.

The results of tri-objective problems, DTLZ1, DTLZ2, and DTLZ7, are presented in Fig. 8–10. Compared with the bi-objective problems, the tri-objective problems can pose a much more challenging task for the selected algorithms. It can be observed that except for AMOCSO, none of the comparison algorithms can approach to the true Pareto front perfectly and they can only form an approximate shape. Moreover, there is a huge performance gap between AMOCSO and the competitors in the distribution of non-dominated solutions, that is, the diversity of AMOCSO is superior to the comparison algorithms on DTLZ1, DTLZ2, and DTLZ7 problems. The statistical experimental data can illustrate the advantages of AMOCSO more clearly compared with those of comparison algorithms. Even on the problem with the smallest IGD performance gap (DTLZ2 problem), the mean IGD result of AMOCSO is still superior to all the competitors over 70%.

Both UF1 and UF4 are bi-objective problems, which are shown in Figs. 11 and 12. Since UF1 is a problem with many local minima, some algorithms can hardly approach to the true Pareto front. Compared with UF1 problem, UF4 problem is easier to converge, and thus all selected algorithms can basicly obtain a non-dominated solution set that is approximate to Pareto front. On the tri-objective problem UF9 where the results are shown in Fig. 13, the performance of the selected comparison algorithms is even worse than on UF1 problem, and only AMOCSO can obtain a satisfactory Pareto front. The performance achieved by CMOPSO and AUDHEIA is inferior to AMOCSO and the remaining comparison algorithms cannot approach to the Pareto front. Some algorithms can achieve excellent performance on individual problems, but cannot maintain that performance on all problems, such as CMOPSO on UF9 and MOEA/D on UF1. The Pareto front obtained by the algorithms can intuitively illustrate their performance in solving MOPs. Experimental results show that the proposed algorithm has a more stable and remarkable performance compared with the comparison algorithms.

4.2.2. Comparisons of the HV metric

IGD metric is calculated by the distance between the obtained non-dominated solution set and the true Pareto front, and thus it mainly reflects the convergence of the algorithms. In order to evaluate the performance of the proposed algorithm comprehensively, the results of HV metric on DTLZ, WFG, and UF benchmark suits are provided in Tables 7 and 8. Similar to the comparison of IGD metric results, several statistics results are shown in tables.

It can be observed that the performance of the proposed AMOCSO algorithm is much more outstanding under the evaluation of HV metric compared with using IGD as the performance metric. There are 20 optimal results obtained by AMOCSO in all 26 test instances. For the 6 problems that AMOCSO suffers setbacks, AMOCSO is slightly inferior to CMOPSO on DTLZ3 problem, to pccsAMOPSO on UF3 problem, and to AUDHEIA on 4 problems (DTLZ6, WFG6, UF2, and UF7 problems). However, all the rankings of AMOCSO are stable in the top several on these 6 problems. Moreover, Wilcoxon's rank-sum test results show that there is no significant difference between the results obtained by the competitors and AMOCSO on 2 of these 6 problems (DTLZ3 and UF7 problems). The experimental results demonstrate that the performance of AMOCSO is promising. On DTLZ and WFG problems, AMOCSO shows perfect performance and accomplishes the best results on all problems except

Table 4
Parameter settings of all the algorithms.

Algorithm	Parameter settings
AMOCSO	$S = 100, N = 100, \phi_0 = 1$
CMOPSO	$S = 100, N = 100, \gamma = 1, p_m = 1/n, \eta_m = 20$
cdMOPSO	$S = 100, N = 100, \omega = 0.4, c_1 = c_2 = 1.429, p_m = 1/n, \eta_m = 20$
pccsAMOPSO	$S = 100, N = 100, \omega(0) = 1, c_1(0) = c_2(0) = 1.429, Step_\omega = 0.00167, Step_{c_1} = Step_{c_2} = 0.0067$
NSGA-II	$S = 100, N = 100, p_m = 1/n, \eta_c = 20, \eta_m = 20$
SPEA2	$S = 100, N = 100, p_m = 1/n, \eta_c = 20, \eta_m = 20$
MOEA/D	$S = 100, N = 100, CR = 1.0, F = 0.5, p_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, \eta_r = 20$
AUDHEIA	$S = 100, N = 100, NA = 20, p_c = 1/n, p_m = 1/n, \eta_c = 20, \eta_m = 20, CR = 1.0, F = 0.5, T = 20, \delta = 0.9,$

for CMOPSO on DTLZ3 and AUDHEIA on DTLZ6 and WFG6. Furthermore, Wilcoxon's rank-sum test results indicate that the results obtained by AMOCSO and CMOPSO are in the same level. On all 10 UF instances, AMOCSO achieves the best performance on 7 problems and the same level as the best result on one of the remaining 3 problems (UF7 problem). The HV performance of AMOCSO is completely dominant on 4 problems (UF1, UF4, UF6, and UF8 problems). Wilcoxon's rank-sum test results show that AMOCSO only lags behind some comparison algorithms on 2 problems (AMOCSO is slightly inferior to AUDHEIA on UF2 problem and to pccsAMOPSO and CMOPSO on UF3 problem).

In general, the proposed algorithm is in a leading position in all problems and has conspicuous advantages over the competitors. The statistical results of ranking and hypothesis testing obviously show the great superiority of AMOCSO. On DTLZ and WFG problems, only AUDHEIA achieves twice better results than AMOCSO and all comparison algorithms have more than 10 times worse results compared to the proposed algorithm. In terms of Rank Sum, the total ranking of AMOCSO is less

Table 5
IGD performance comparison on DTLZ and WFG instances (all results are averaged on 30 independent runs).

Problem	Index	MOCSOs		MOPSOs		MOEAs			
		AMOCSO	CMOPSO	cdMOPSO	pccsAMOPSO	NSGA-II	SPEA2	MOEA/D	AUDHEIA
DTLZ1	Mean	**3.43e−3**	4.43e−2	2.75e−1	1.36e−1	1.42e−1	3.77e−1	6.04e−1	1.01e−2
	Std.	**2.76e−4**	7.83e−2	9.34e−2	1.12e−1	6.67e−3	1.45e−4	2.89e−1	1.75e−4
	Rank	**1**	3−	6−	4−	5−	7−	8−	2−
DTLZ2	Mean	**1.09e−3**	4.33e−3	1.02e−1	6.14e−2	1.06e−1	8.22e−2	6.24e−1	6.68e−3
	Std.	**9.85e−5**	5.84e−4	1.34e−3	1.90e−3	8.38e−3	2.83e−3	6.44e−5	4.06e−4
	Rank	**1**	2≈	6−	4−	7−	5−	8−	3−
DTLZ3	Mean	**2.44e−2**	1.23e−1	4.46e−1	3.19e−1	6.64e−1	4.88e−1	6.52e−1	2.85e−2
	Std.	**3.31e−4**	9.68e−2	1.02e−1	2.86e−1	7.57e−2	1.00e−2	2.95e−1	5.21e−4
	Rank	**1**	3−	7−	4−	6−	8−	5−	2≈
DTLZ4	Mean	**1.25e−3**	4.41e−3	1.02e−1	4.41e−2	7.31e−2	7.29e−2	2.70e−1	2.92e−2
	Std.	**6.86e−4**	7.58e−4	3.66e−2	3.37e−3	5.09e−2	1.42e−3	6.84e−3	2.02e−3
	Rank	**1**	2−	7−	4−	6−	5−	8−	3−
DTLZ5	Mean	9.06e−4	4.32e−3	6.05e−3	6.62e−3	8.05e−3	1.41e−2	5.94e−1	**8.12e−4**
	Std.	2.25e−5	3.97e−5	7.29e−4	5.56e−4	1.63e−3	3.54e−3	8.43e−5	**3.01e−5**
	Rank	2	3−	4−	5−	6−	7−	8−	**1+**
DTLZ6	Mean	8.70e−4	4.16e−3	5.23e−3	5.04e−3	1.47e+0	2.50e−1	6.17e−1	**7.85e−4**
	Std.	1.03e−5	2.49e−5	3.90e−4	2.28e−4	6.09e−1	5.67e−2	1.01e−4	**4.17e−5**
	Rank	2	3−	5−	4−	8−	6−	7−	**1+**
DTLZ7	Mean	**5.86e−3**	4.43e−2	2.99e−2	4.28e−2	6.14e−1	6.21e−2	6.57e−1	2.98e−2
	Std.	**6.43e−4**	4.43e−5	1.65e−3	9.51e−4	1.16e−1	0.00e+0	9.87e−4	9.56e4
	Rank	**1**	5−	3−	4−	6−	7−	8−	2−
WFG1	Mean	**3.83e−3**	9.62e−3	9.86e−1	8.36e−3	6.29e−1	1.22e+0	1.60e−2	5.81e−3
	Std.	**9.25e−4**	6.74e−2	3.42e−2	4.36e−4	2.18e−1	2.07e−1	7.81e−3	1.36e−4
	Rank	**1**	4−	7−	3−	6−	8−	5−	2−
WFG2	Mean	**7.07e−3**	1.12e−2	8.62e−2	2.57e−2	1.04e−1	9.33e−2	4.44e−2	1.26e−2
	Std.	**5.47e−4**	2.33e−4	1.04e−2	6.90e−3	7.35e−2	6.53e−2	2.61e−2	6.89e−3
	Rank	**1**	2−	6−	4−	8−	7−	5−	3−
WFG3	Mean	**5.05e−3**	6.32e−3	9.22e−3	5.83e−3	7.40e−3	6.07e−3	6.73e−3	5.61e−3
	Std.	**4.14e−5**	2.49e−4	5.45e−4	3.32e−4	3.39e−4	4.01e−4	6.41e−6	3.76e−4
	Rank	**1**	5≈	8−	3≈	7−	4≈	6≈	2≈
WFG4	Mean	**5.01e−3**	5.88e−3	7.21e−3	5.28e−3	8.07e−3	6.26e−3	6.90e−3	5.25e−3
	Std.	**8.87e−5**	7.95e−4	9.03e−4	2.12e−4	1.33e−3	1.76e−4	3.91e−4	1.01e−4
	Rank	**1**	4≈	7−	3≈	8−	5−	6−	2≈
WFG5	Mean	**2.11e−2**	2.35e−2	6.53e−2	6.50e−2	6.54e−2	6.49e−2	6.45e−2	6.35e−2
	Std.	**6.36e−4**	2.49e−3	8.85e−4	3.19e−3	3.07e−3	3.19e−3	1.68e−4	3.19e−3
	Rank	**1**	2≈	5−	4−	6−	8−	7−	3−
WFG6	Mean	6.99e−3	1.01e−2	7.30e−3	7.07e−3	8.64e−3	1.27e−2	7.59e−3	**6.75e−3**
	Std.	2.02e−3	7.72e−3	1.28e−3	1.17e−3	9.77e−4	6.86e−3	4.01e−3	**1.03e−3**
	Rank	2	7−	4≈	3≈	6	8−	5−	**1**≈
WFG7	Mean	**1.71e−3**	2.26e−2	5.78e−3	6.20e−3	8.72e−3	8.82e−3	7.18e−3	5.93e−3
	Std.	**5.46e−4**	3.20e−3	8.73e−5	1.18e4	4.79e−4	6.33e−4	5.52e−4	9.65e−5
	Rank	**1**	8−	2−	4−	6−	7−	5−	3−
WFG8	Mean	3.52e−3	**3.32e−3**	2.70e−2	7.89e−3	2.26e−2	5.13e−2	2.04e−2	6.63e−3
	Std.	4.77e−4	**5.53e−4**	3.65e−3	4.25e−4	3.28e−3	7.98e−3	2.29e−3	3.96e−4
	Rank	2	**1+**	7−	4−	6−	8−	5−	3−
WFG9	Mean	**4.49e−3**	2.17e−2	3.86e−2	6.17e−3	7.96e−3	6.45e−3	7.48e−3	6.21e−3
	Std.	**9.58e−5**	2.24e−3	7.83e−3	1.68e−4	3.89e−4	2.91e−4	3.73e−4	1.02e−4
	Rank	**1**	7−	8−	2−	6−	4−	5−	3−
Rank Sum		**20**	61	92	59	103	104	101	36
Final Rank		**1**	4	5	3	7	8	6	2
+/ − / ≈		−	1/12/3	0/15/1	0/13/3	0/16/0	0/15/1	0/15/1	2/10/4

"+", "−" and "≈" represent that the results of the selected algorithms are superior, inferior, and similar to those obtained by AMOCSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value = 0.05.

Table 6
IGD performance comparison on UF instances (all results are averaged on 30 independent runs).

Problem	Index	MOCSOs		MOPSOs		MOEAs			
		AMOCSO	CMOPSO	cdMOPSO	pccsAMOPSO	NSGA-II	SPEA2	MOEA/D	AUDHEIA
UF1	Mean	3.07e−3	3.13e−2	1.93e−1	5.82e−e−2	7.03e−2	9.02e−2	**1.79e−3**	2.53e−3
	Std.	1.42e−4	5.21e−3	9.38e−3	1.52e−2	1.14e−2	1.45e−2	**1.93e−4**	1.07e−4
	Rank	3	4−	8−	5−	6−	7−	**1+**	2≈
UF2	Mean	**3.26e−3**	1.71e−2	2.23e−2	2.19e−2	2.34e−2	2.55e−2	6.53e−3	5.81e−3
	Std.	**2.24e−5**	4.24e−3	2.26e−3	9.00e−3	6.66e−3	6.15e−3	1.74e−3	4.98e−4
	Rank	**1**	4−	6−	5−	7−	8−	3−	2−
UF3	Mean	**4.04e−2**	6.13e−2	1.18e−1	1.31e−1	1.17e−1	1.61e−1	9.13e−2	7.28e−2
	Std.	**5.91e−3**	1.23e−2	3.81e−2	3.24e−2	2.93e−2	4.29e−2	1.32e−2	6.10e−3
	Rank	**1**	2−	6−	7−	5−	8−	4−	3−
UF4	Mean	3.85e−2	6.59e−2	5.86e−2	3.83e−2	5.09e−2	5.12e−2	6.39e−2	**3.77e−2**
	Std.	6.12e−4	3.44e−3	2.73e−3	1.02e−3	5.92e−4	6.70e−4	4.96e−3	**7.49e−4**
	Rank	3	8−	6−	2≈	4−	5−	7−	**1**≈
UF5	Mean	**1.64e−1**	3.46e−1	1.17e+0	2.18e−1	2.23e−1	2.32e−1	2.95e−1	2.05e−1
	Std.	**3.35e−2**	1.76e−1	4.91e−1	5.16e−2	4.22e−2	5.43e−2	1.15e−1	1.17e−1
	Rank	**1**	7−	8−	3−	4−	5−	6−	2−
UF6	Mean	1.45e−1	1.50e−1	4.41e−1	**1.14e−1**	1.17e−1	1.37e−1	1.95e−1	1.42e−1
	Std.	6.06e−2	1.08e−1	6.38e−2	**2.19e−2**	3.52e−2	6.37e−2	1.73e−1	8.47e−2
	Rank	5	6≈	8−	**1+**	2+	3≈	7−	4≈
UF7	Mean	**2.73e−3**	3.21e−2	2.15e−2	4.84e−2	6.51e−2	9.51e−2	2.95e−3	3.09e−3
	Std.	**1.47e−4**	6.09e−2	2.54e−3	7.81e−2	8.31e−2	1.18e−1	2.57e−4	5.56e−4
	Rank	**1**	5−	4−	6−	7−	8−	2≈	3≈
UF8	Mean	5.72e−1	5.08e−1	9.86e−1	9.66e−2	6.92e−1	1.64e−0	5.81e−1	**4.56e−1**
	Std.	9.46e−3	6.67e−2	3.42e−2	4.08e−2	1.28e−1	2.07e−1	1.36e−2	**8.18e−2**
	Rank	3	2≈	7−	6−	5−	8−	4≈	**1+**
UF9	Mean	**9.96e−2**	1.04e−1	8.62e−1	1.67e−1	7.06e−1	9.33e−1	3.14e−1	1.26e−1
	Std.	**6.36e−2**	7.35e−2	1.04e−1	7.13e−2	1.41e−1	6.53e−1	2.61e−2	6.89e−2
	Rank	**1**	2≈	7−	4−	6−	8−	5−	3≈
UF10	Mean	**2.04e−1**	3.44e+0	9.22e−1	2.95e−1	7.40e−1	6.07e+0	4.68e−1	3.16e−1
	Std.	**3.03e−2**	3.76e−1	5.45e−1	2.04e−2	3.39e−2	4.01e−1	1.14e−1	4.51e−1
	Rank	**1**	7−	6−	2≈	5−	8−	4−	3−
Rank Sum		**20**	47	66	41	51	68	43	24
Final Rank		**1**	5	7	3	6	8	4	2
+/ − / ≈		−	0/7/3	0/10/0	1/7/2	1/9/0	0/9/1	1/7/2	1/4/5

"+", "−" and "≈" represent that the results of the selected algorithms are superior, inferior, and similar to those obtained by AMOCSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value = 0.05.

than 20, which is much better than the second-ranked algorithm (AUDHEIA obtains 37). On UF problems, AMOCSO also has considerable advantages over the comparison algorithms. It can be observed from the statistical results that the overall ranking of AMOCSO on all UF problems is just 16, while the best performed competitor (AUDHEIA) is up to 26. The comprehensive scores of the selected algorithms in solving all benchmark problems can be obtained in the rows of Final Rank. Four state-of-the-art algorithms, AMOCSO, AUDHEIA, CMOPSO, and pccsAMOPSO, obtain the top four rankings sequentially, which are also the top four contestants in the IGD metric results.

HV metric can reflect the comprehensive performance of the algorithm, that is, both the convergence and the diversity of the algorithm. Therefore, the plotted figures of the non-dominated solution set can intuitively demonstrate the advantages and the disadvantages of the slected algorithms. The obvious comparison of the comprehensive performance can be observed in Fig. 4–13. For DTLZ2 problem plotted in Fig. 9, it is a typical problem to observe the comprehensive performance gap between AMOCSO and the comparison algorithms. It can be seen clearly that the four classical algorithms, cdMOPSO, NSGA-II, SPEA2, and MOEA/D, have distinct defects in converging to the true Pareto front. Compared with the above classical algorithms, CMOPSO and AUDHEIA have achieved quite well performance. However, there is a visible performance disparity in the distribution uniformity of non-dominated solutions between CMOPSO, AUDHEIA, and AMOCSO. Specifically, the HV metric results of the remaining three comparison algorithms, CMOPSO, pccsAMOPSO, and AUDHEIA, are at a similar level, while the experimental data of the proposed AMOCSO is significantly better than them. The HV metric results of AMOCSO are over 70% better than the competitors on DTLZ2 problem. Performance gap is more obvious on UF problems. cdMOPSO cannot approach to the true Pareto front, and the non-dominated solutions obtained by NSGA-II and SPEA2 cannot completely cover the whole Pareto front. Moreover, the better performed algorithms, CMOPSO and pccsAMOPSO, also have the problem of incomplete coverage on the Pareto front. The results of HV metric are different from that of IGD metric and the numerical gap is not particularly obvious, but AMOCSO is still more than 10% ahead of the comparison algorithms on the problem with minimum performance gap (UF4 problem).

4.2.3. Comprehensive comparison

In order to clearly demonstrate the overall performance of the proposed AMOCSO and the selected comparison algorithms on all instances, the summarized results are presented in Table 9.

Table 7
HV performance comparison on DTLZ and WFG instances (all results are averaged on 30 independent runs).

Problem	Index	MOCSOs		MOPSOs		MOEAs			
		AMOCSO	CMOPSO	cdMOPSO	pccsAMOPSO	NSGA-II	SPEA2	MOEA/D	AUDHEIA
DTLZ1	Mean	**1.18e+0**	9.53e−1	7.53e−1	7.66e−1	8.87e−1	6.82e−1	6.75e−1	9.21e−1
	Std.	**6.36e−3**	2.10e−3	4.53e−2	3.78e−3	3.27e−2	4.49e−1	7.33e−1	3.97e−3
	Rank	**1**	2−	6−	5−	4−	7−	8−	3−
DTLZ2	Mean	**1.77e+0**	9.26e−1	3.54e−1	4.07e−1	3.39e−1	3.80e−1	2.85e−1	9.98e−1
	Std.	**4.61e−3**	5.03e−3	5.61e−3	2.81e−4	9.60e−3	3.54e−3	1.88e−2	1.09e−3
	Rank	**1**	3−	6−	4−	7−	5−	8−	2−
DTLZ3	Mean	1.02e+0	**1.09e+0**	0.00e+0	8.23e−1	0.00e+0	0.00e+0	9.79e−1	9.87e−1
	Std.	8.38e−4	**3.72e−3**	0.00e+0	6.74e−2	0.00e+0	0.00e+0	5.54e−2	2.56e−2
	Rank	2	**1**≈	6−	5−	7−	8−	4−	3≈
DTLZ4	Mean	**9.62e−1**	7.85e−1	2.05e−1	4.62e−1	3.43e−1	2.04e−1	2.79e−1	5.27e−1
	Std.	**6.01e−3**	6.49e−3	4.49e−2	1.96e−1	9.80e−3	1.65e−2	4.76−2	2.96e−2
	Rank	**1**	2−	7−	4−	5−	8−	6−	3−
DTLZ5	Mean	**4.47e−1**	3.68e−1	9.20e−2	3.31e−1	8.98e−2	9.07e−2	7.10e−2	4.24e−1
	Std.	**3.12e−3**	1.87e−4	1.37e−4	1.48e−4	4.21e−4	3.32e−4	7.34e−4	6.08e−3
	Rank	**1**	3−	5−	4−	7−	6−	8−	2≈
DTLZ6	Mean	5.25e−1	5.24e−1	0.00e+0	3.07e−2	0.00e+0	0.00e+0	7.06e−2	**6.03e−1**
	Std.	1.01e−3	3.21e−3	0.00e+0	2.61e−2	0.00e+0	0.00e+0	7.74e−6	**8.42e−3**
	Rank	2	3≈	6−	5−	7−	8−	4−	**1+**
DTLZ7	Mean	**7.22e−1**	5.12e−1	1.86e−1	2.62e−1	2.53e−1	2.68e−1	8.95e−2	5.43e−1
	Std.	**2.94e−3**	1.04e−2	1.34e−2	9.44e−3	8.17e−3	1.53e−2	1.07e−1	9.72e−3
	Rank	**1**	3−	7−	5−	6−	4−	8−	2−
WFG1	Mean	**9.15e−1**	6.56e−1	2.75e−1	3.06e−1	3.46e−1	2.94e−1	1.09e−1	7.13e−1
	Std.	**5.44e−2**	5.12e−2	1.26e−1	8.06e−2	1.92e−1	1.77e−1	3.83e−3	4.03e−4
	Rank	**1**	3−	7−	5−	4−	6−	8−	2−
WFG2	Mean	**8.11e−1**	7.12e−1	4.12e−1	5.64e−1	5.61e−1	5.69e−1	5.44e−1	6.06e−1
	Std.	**6.46e−3**	4.77e−3	8.18e−3	1.98e−3	1.27e−3	8.66e−4	4.20e−3	2.32e−3
	Rank	**1**	2−	8−	5−	6−	4−	7−	3−
WFG3	Mean	**8.19e−1**	5.38e−1	4.17e−1	4.39e−1	4.36e−1	4.38e−1	4.37e−1	6.31e−1
	Std.	**7.30e−4**	4.93e−4	2.64e−2	2.79e−4	1.05e−3	5.56e−4	5.01e−4	1.40e−4
	Rank	**1**	3−	8−	4−	7−	5−	6−	2−
WFG4	Mean	**5.25e−1**	3.15e−1	1.94e−1	2.16e−1	2.11e−1	2.13e−1	1.87e−1	3.91e−1
	Std.	**4.69e−3**	4.01e−3	8.45e−3	2.79e−4	1.32e−3	5.46e−4	3.70e−3	2.02e−3
	Rank	**1**	3−	7−	4−	6−	5−	8−	2−
WFG5	Mean	**6.02e−1**	1.92e−1	7.38e−2	1.93e−1	1.89e−1	1.91e−1	1.88e−1	2.02e−1
	Std.	**9.19e−4**	4.23e−3	1.73e−4	7.95e−5	1.49e−3	4.22e−4	5.51e−4	3.24e−3
	Rank	**1**	3−	8−	4−	6−	5−	7−	2−
WFG6	Mean	2.26e−1	2.00e−1	1.92e−1	1.91e−1	1.90e−1	1.92e−1	1.97e−1	**4.57e−1**
	Std.	4.73e−3	6.45e−2	2.68e−2	7.27e−3	1.35e−2	2.34e−2	1.59e−3	**2.95e−3**
	Rank	2	3≈	6≈	7−	8−	5≈	4≈	**1+**
WFG7	Mean	**4.79e−1**	3.93e−1	2.06e−1	2.07e−1	2.03e−1	2.05e−1	1.98e−1	3.91e−1
	Std.	**7.10e−4**	9.51e−3	1.26e−3	6.48e−5	9.87e−4	8.22e−4	6.97e−4	1.52e−3
	Rank	**1**	2≈	5−	4−	7−	6−	8−	3≈
WFG8	Mean	**4.27e−1**	1.74e−1	1.41e−1	1.46e−1	1.40e−1	1.43e−1	1.39e−1	1.68e−1
	Std.	**2.13e−3**	7.96e−3	6.15e−3	2.06e−3	2.56e−3	1.26e−3	3.09e−3	1.59e−3
	Rank	**1**	2−	6−	4−	7−	5−	8−	3−
WFG9	Mean	**4.55e−1**	4.32e−1	2.17e−1	2.36e−1	2.27e−1	2.30e−1	5.25e−2	3.38e−1
	Std.	**6.34e−4**	9.06e−4	4.07e−3	1.72e−3	3.48e−3	3.11e−3	1.04e−3	2.10e−3
	Rank	**1**	2≈	7−	4−	6−	5−	8−	3−
Rank Sum		**19**	40	105	73	100	92	110	37
Final Rank		**1**	3	8	4	6	5	7	2
+/ − / ≈		—	0/11/5	0/15/1	0/16/0	0/16/0	0/15/1	0/15/1	2/11/3

"+", "−" and "≈" represent that the results of the selected algorithms are superior, inferior, and similar to those obtained by AMOCSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value = 0.05.

It can be observed that the number of worse results ("−") obtained by the comparison algorithms is at a high level. Particularly, five comparison algorithms, cdMOPSO, pccsAMOPSO, NSGA-II, SPEA2, and MOEA/D, obtain more than 40 worse results. AUDHEIA and CMOPSO are the only two comparison algorithms with 30 worse results, and they are also the algorithms that can obtain 10 similar results ("≈"). pccsAMOPSO has performed equally well and it acquires 2 better results ("+") and 7 similar results. These three algorithms, AUDHEIA, CMOPSO, and pccsAMOPSO, occupy the first, the second, and the third place respectively in all competitors. However, the results of the top three competitors are still far worse than the results obtained by AMOCSO, and the number of worse results obtained by them is far greater than the sum of better and similar results. Some comparison algorithms perform very well on specific problems with one performance metric, such as AUDHEIA on DTLZs and UFs with IGD results, but they are embarrassing on the other problems and performance metric results, which leads to the final experimental results that their overall performance is far inferior to the proposed algorithm.

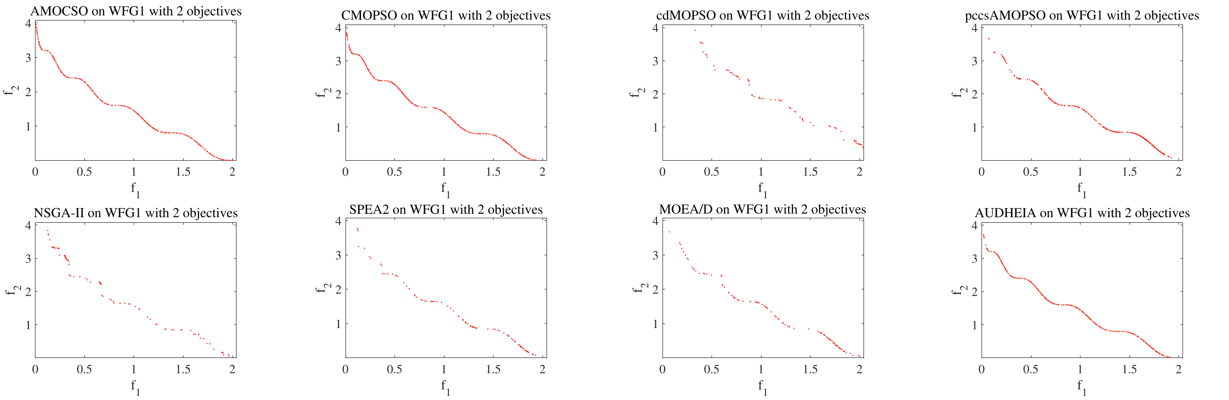

Fig. 4. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on bi-objective WFG1.

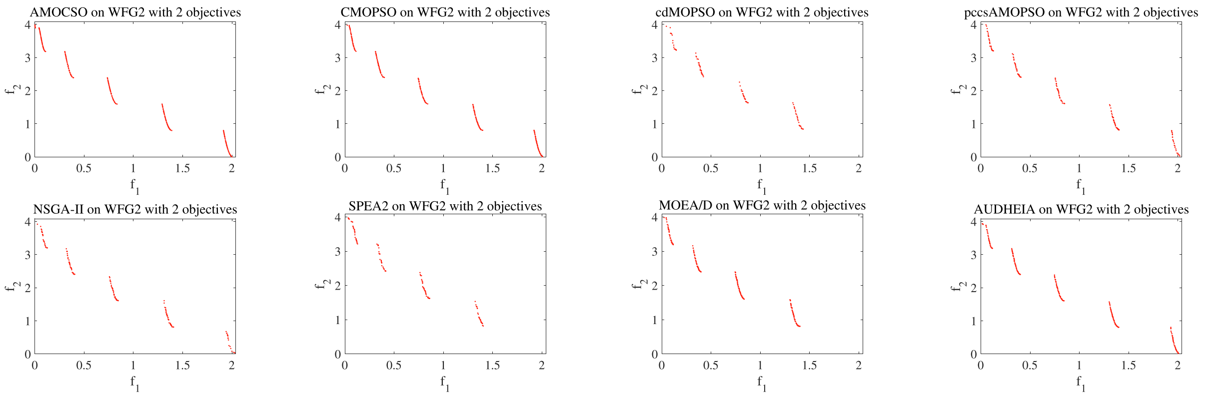

Fig. 5. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on bi-objective WFG2.

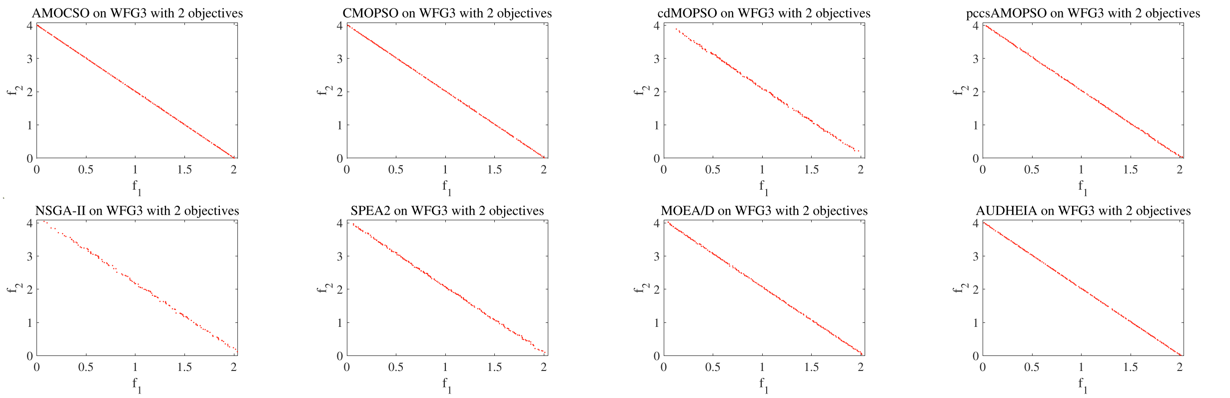

Fig. 6. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on bi-objective WFG3.

Based on the above discussion and analysis, the proposed AMOCSO algorithm not only performs commendably in convergence, but also works well in diversity. The experimental results of HV metric show that AMOCSO has a prominent comprehensive performance. It is worth noting that the stable performance of the algorithms is closely related to the artificial parameters contained in them. The comparison algorithms require extensive manual parameter settings, and due to the variation of the solution space for different problems, the fixed parameter settings cannot give full play to the performance of the

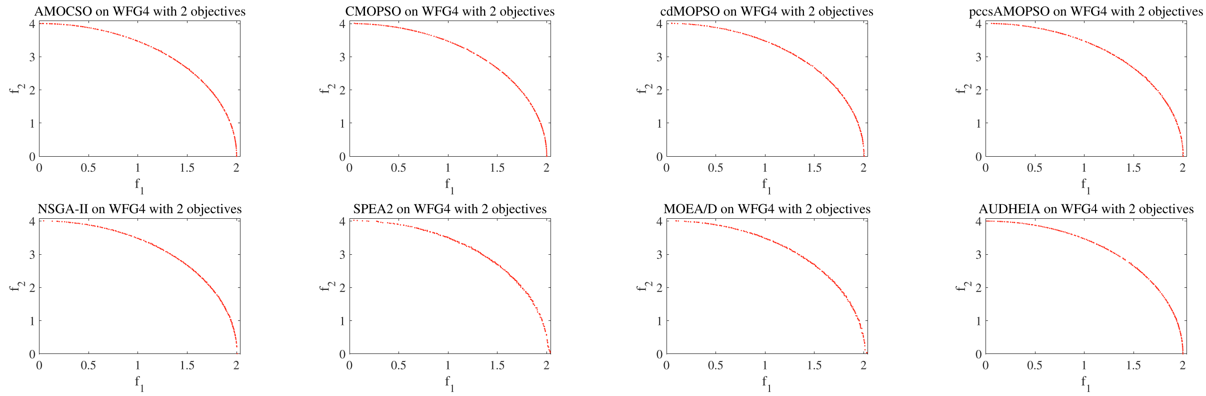

Fig. 7. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on bi-objective WFG4.

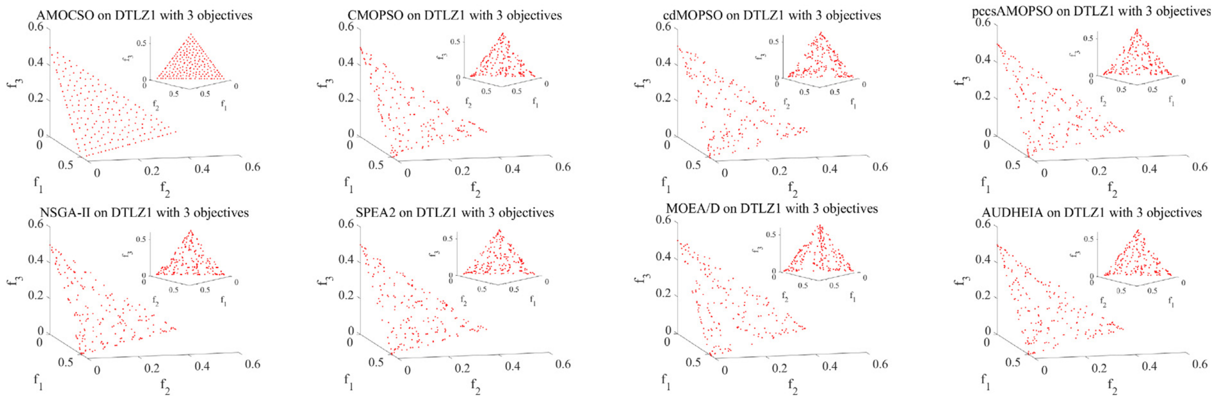

Fig. 8. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective DTLZ1.

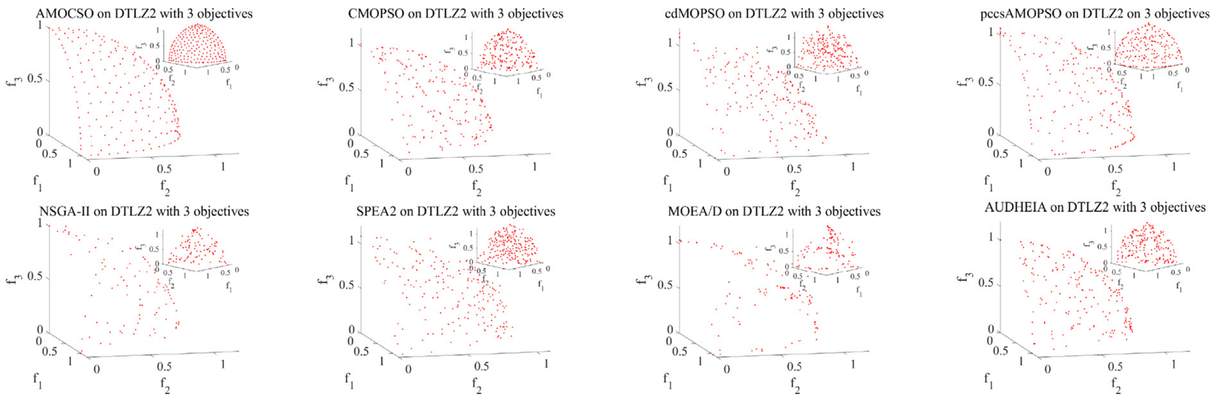

Fig. 9. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective DTLZ2.

algorithms. Therefore, the outstanding performance of the proposed algorithm reflects its promising generalization, which is one of the reasons that the proposed algorithm can show much better performance and stability than the comparison algorithms on all test problems.

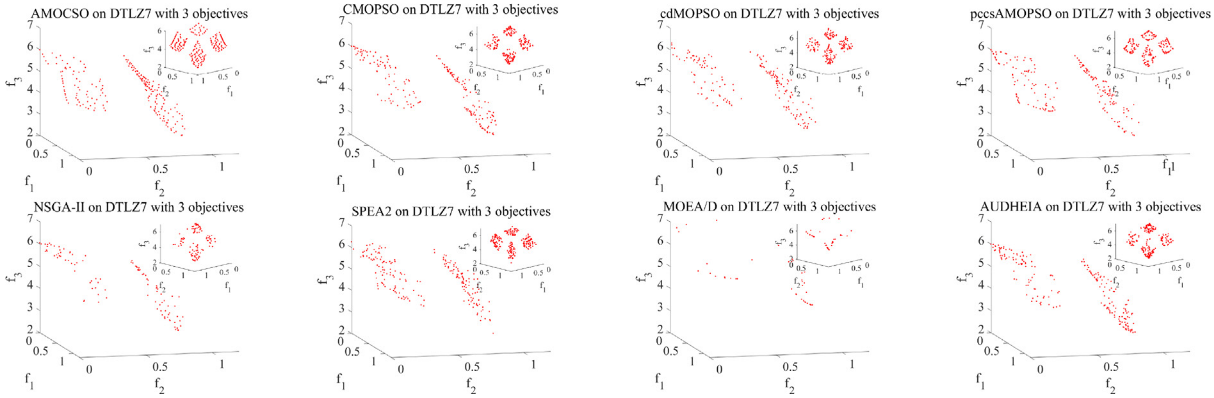

Fig. 10. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective DTLZ7.

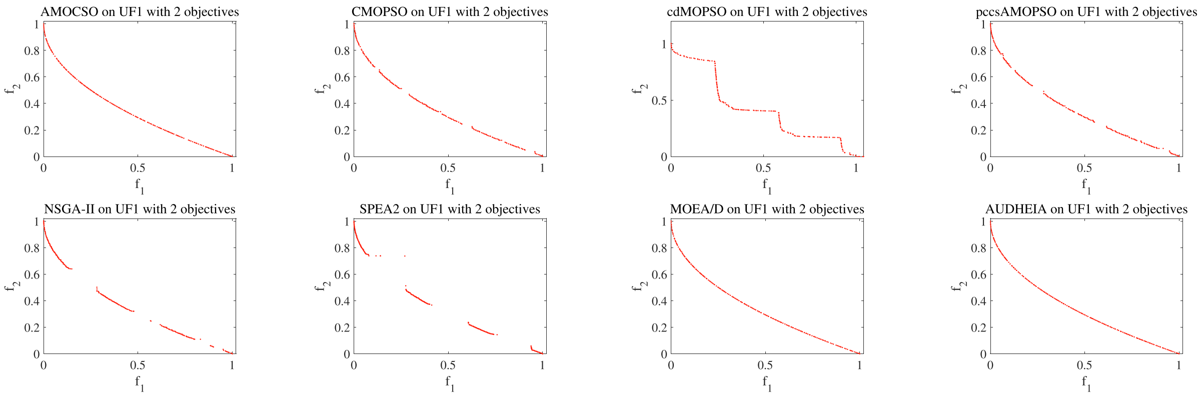

Fig. 11. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective UF1.

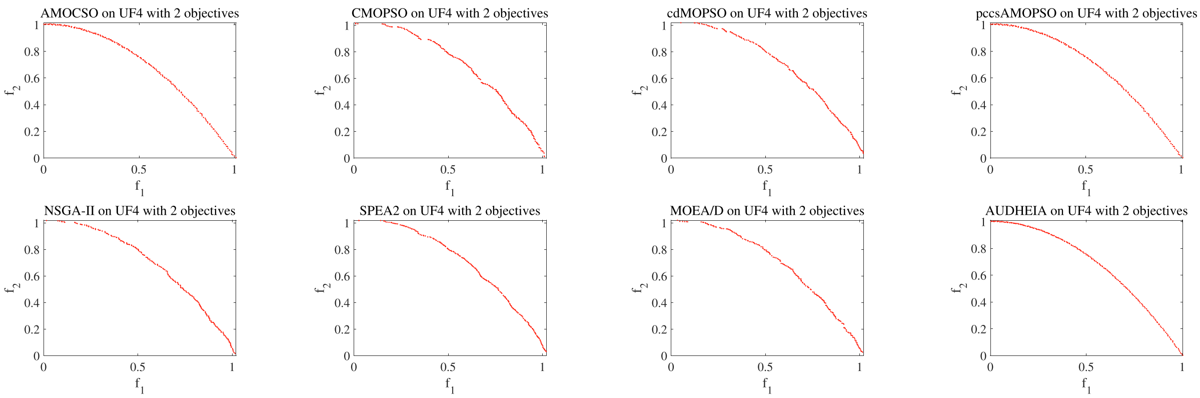

Fig. 12. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective UF4.

5. Conclusions

A novel multi-objective optimization algorithm, adaptive multi-objective competitive swarm optimizer (AMOCSO), is proposed in this paper. Abundant experiments are carried out and several state-of-the-art algorithms are set as the competitors. The experimental results demonstrate that the proposed algorithm has favorable convergence and diversity. It can be

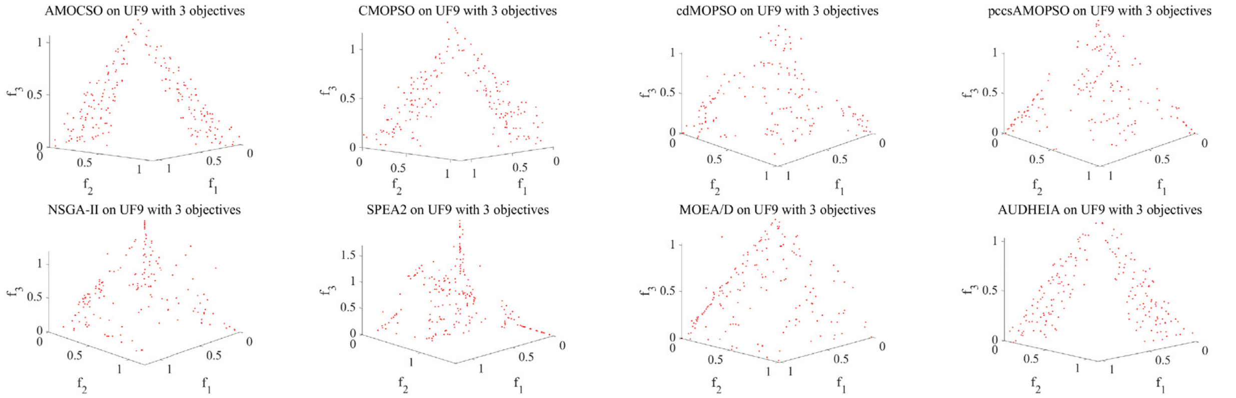

Fig. 13. Non-dominated solutions obtained by AMOCSO, cdMOPSO, pccsAMOPSO, CMOPSO, NSGA-II, SPEA2, MOEA/D, and AUDHEIA on tri-objective UF9.

easily observed from the intuitive results that the non-dominated solutions obtained by AMOCSO can perfectly approximate the true Pareto front and distribute uniformly. Compared with other multi-objective optimization algorithms, the main advantages of the proposed AMOCSO algorithm can be summarized as follows.

1. An advanced optimization scheme, competitive mechanism, is the basic instruction of AMOCSO. It can stimulate the potentiality of the particles, and greatly improves the global exploration capability of the proposed algorithm. The best embodiment of the excellent global exploration ability is that AMOCSO does not even need a mutation operator to prevent it from falling into local optimum in solving complex MOPs.

Table 8
HV performance comparison on UF instances (all results are averaged on 30 independent runs).

Problem	Index	MOCSOs		MOPSOs		MOEAs			
		AMOCSO	CMOPSO	cdMOPSO	pccsAMOPSO	NSGA-II	SPEA2	MOEA/D	AUDHEIA
UF1	Mean	**9.27e−1**	6.74e−1	1.23e−1	6.75e−1	1.19e−1	1.14e−1	8.79e−1	8.82e−1
	Std.	**6.32e−3**	4.10e−2	8.29e−2	9.41e−2	8.03e−2	8.34e−2	3.82e−2	1.07e−2
	Rank	**1**	4−	6−	5−	8−	7−	3−	2−
UF2	Mean	1.06e+0	9.22e−1	6.80e−1	9.47e−1	7.21e−1	6.83e−1	7.72e−1	**1.60e+0**
	Std.	1.03e−2	3.13e−2	4.74e−2	6.56e−2	5.52e−2	6.07e−2	6.01e−2	**3.87e−2**
	Rank	2	4−	8−	3≈	6−	7−	5−	**1+**
UF3	Mean	9.03e−1	9.19e−1	8.33e−1	**9.29e−1**	8.42e−1	8.55e−1	8.02e−1	9.12e−1
	Std.	4.80e−2	3.21e+0	2.73e−1	**2.13e−2**	3.04e−2	5.79e−2	6.21e−2	5.09e−2
	Rank	4	2+	7−	**1+**	6−	5−	8−	3≈
UF4	Mean	**6.63e−1**	5.01e−1	2.64e−1	6.02e−1	3.18e−1	4.21e−1	5.64e−1	4.47e−1
	Std.	**2.84e−2**	7.03e−2	4.95e−1	5.43e−2	4.83e−2	8.06e−2	3.85e−2	6.38e−2
	Rank	**1**	4−	8−	2−	7−	6−	3−	5−
UF5	Mean	**1.99e−1**	1.94e−1	1.89e−1	1.95e−1	0.00e+0	1.88e−1	1.91e−1	1.90e−1
	Std.	**9.78e−2**	1.25e−1	1.27e−1	1.33e−1	0.00e+0	1.22e−1	1.14e−1	1.25e−1
	Rank	**1**	3≈	6−	2≈	8−	7−	4≈	5−
UF6	Mean	**2.96e−1**	2.34e−1	2.36e−1	2.67e−1	2.03e−1	2.00e−1	2.38e−1	2.84e−1
	Std.	**4.95e−2**	8.33e−2	7.95e−2	6.43e−2	6.96e−2	7.21e−1	7.02e−2	6.32e−2
	Rank	**1**	6−	5−	3−	7−	8−	4−	2−
UF7	Mean	6.52e−1	6.17e−1	5.71e−1	6.23e−1	6.08e−1	6.02e−1	6.53e−1	**6.78e−1**
	Std.	2.36e−2	4.22e−2	9.46e−2	5.83e−2	7.03e−2	5.29e−2	3.84e−2	**1.47e−2**
	Rank	3	5−	8−	4−	6−	7−	2≈	**1≈**
UF8	Mean	**6.83e−1**	6.71e−1	5.49e−1	5.82e−1	5.45e−1	5.42e−1	5.60e−1	5.94e−1
	Std.	**3.35e−2**	4.93e−2	9.36e−2	5.04e−2	7.34e−2	7.99e−2	8.87e−2	7.37e−2
	Rank	**1**	2−	6−	4−	7−	8−	5−	3−
UF9	Mean	**9.18e−1**	8.97e−1	8.80e−1	9.02e−1	8.95e−1	8.84e−1	8.92e−1	9.14e−1
	Std.	**2.73e−2**	5.94e−2	3.49e−2	5.36e−2	4.27e−2	4.70e−2	5.91e−2	3.46e−2
	Rank	**1**	4≈	8−	3−	5−	7−	6−	2≈
UF10	Mean	**1.85e−1**	1.79e−1	1.26e−1	1.75e−1	1.24e−1	1.11e−1	1.74e−1	1.80e−1
	Std.	**4.22e−3**	5.82e−3	3.37e−2	4.64e−2	3.54e−2	6.06e−2	4.39e−2	4.68e−2
	Rank	**1**	3≈	6−	4−	7−	8−	5−	2−
Rank Sum		**16**	37	68	31	67	70	45	26
Final Rank		**1**	4	7	3	6	8	5	2
+/−/≈		—	0/7/3	0/10/0	1/7/2	0/10/0	0/10/0	0/8/2	1/6/3

"+", "−" and "≈" represent that the results of the selected algorithms are superior, inferior, and similar to those obtained by AMOCSO respectively, which is judged by Wilcoxon's rank-sum test with the p-value = 0.05.

Table 9
Overall performance comparison of the algorithms on all instances.

Problem	Index		Algorithms							
			AMOCSO	CMOPSO	cdMOPSO	pccsAMOPSO	NSGA-II	SPEA2	MOEA/D	AUDHEIA
DTLZs	IGD	Rank	**9**	21	38	29	44	43	52	14
		$+/-/\approx$	–	0/6/1	0/7/0	0/7/0	0/7/0	0/7/0	0/7/0	2/4/1
	HV	Rank	**9**	17	43	32	43	46	46	16
		$+/-/\approx$	–	0/5/2	0/7/0	0/7/0	0/7/0	0/7/0	0/7/0	1/4/2
WFGs	IGD	Rank	**11**	40	54	30	59	61	49	22
		$+/-/\approx$	–	1/6/2	0/8/1	0/6/3	0/9/0	0/8/1	0/8/1	0/6/3
	HV	Rank	**10**	23	62	41	57	46	64	21
		$+/-/\approx$	–	0/6/3	0/8/1	0/9/0	0/9/0	0/8/1	0/8/1	1/7/1
UFs	IGD	Rank	**20**	47	66	41	51	68	43	24
		$+/-/\approx$	–	0/7/3	0/10/0	1/7/2	1/9/0	0/9/1	1/7/2	1/4/5
	HV	Rank	**16**	37	68	31	67	70	45	26
		$+/-/\approx$	–	0/7/3	0/10/0	1/7/2	0/10/0	0/10/0	0/8/2	1/6/3
Rank Sum			**75**	185	331	204	321	334	299	123
Final Rank			**1**	3	7	4	6	8	5	2
Total better/worse/similar			–	1/37/14	0/50/2	2/43/7	1/51/0	0/49/3	1/45/6	6/31/15
Final result (vs AMOCSO)			–	Worse	Worse	Worse	Worse	Worse	Worse	Worse

2. A balanced optimization scheme is designed to coordinate the improved competitive mechanism. The winners and the losers obtained in competition are guided respectively to promote the convergence and the diversity. Individuals in the population are assigned different goals, which significantly enhance the optimization efficiency of the proposed algorithm.
3. A crucial problem universally existed in metaheuristic algorithms, the parameter settings, has been solved thoroughly in AMOCSO. A feasible and practical adaptive strategy is proposed to detect the evolutionary environment and formulate the scheme of parameter adjustment. The human factors has been decreased to the minimum to promote the generalization of AMOCSO. With the processing of parameters, the evolutionary states of the proposed algorithm can be switched freely as needed.

In this paper, the proposed algorithm is improved from the perspective of synthesis, and it is of great significance that keeps on studying the multi-objective algorithm based on the competitive mechanism (MOCSO). Since the brief and efficient optimization process is the most brilliant characteristic of MOCSO, it can also be employed to solve the large-scale MOPs and many-objective optimization problems (MaOPs), which are brand-new topics. Therefore, we will continue to focus on the research of MOCSO and try to solve some real-world problems using the proposed algorithm.

CRediT authorship contribution statement

Weimin Huang: Conceptualization, Methodology, Software, Data-curation, Writing-original-draft. **Wei Zhang:** Visualization, Supervision, Validation, Funding-acquisition, Writing-review-editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by National Natural Science Foundation of China under Grants [61703145]; Scientific and technological innovation team of colleges and universities in Henan Province under Grants [20IRTSTHN019]. In addition, we are grateful to the anonymous reviewers and editors for their valuable suggestions and comments on our manuscript.

References

[1] W. Peng, Q. Zhang, A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems, IEEE International Conference on Granular Computing 2008 (2008) 534–537.
[2] M. Pescador-Rojas, C.A.C. Coello, Collaborative and adaptive strategies of different scalarizing functions in moea/d, IEEE Cong. Evol. Comput. (CEC) 2018 (2018) 1–8, https://doi.org/10.1109/CEC.2018.8477815.
[3] Y. Hu, Y. Zhang, D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, IEEE Trans. Cybern. 51 (2021) 874–888.
[4] N.A. Moubayed, A. Petrovski, J. Mccall, D2mopso: Mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces, Evol. Comput. 22 (2014) 47–77.
[5] M. Li, S. Yang, X. Liu, Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization, IEEE Transactions on Evolutionary Computation 20 (2016) 645–665.

[6] W. Feng, D. Gong, Z. Yu, Multi-objective evolutionary optimization based on online perceiving pareto front characteristics, Information Sciences 581 (2021) 912–931.
[7] W. Huang, W. Zhang, Adaptive multi-objective particle swarm optimization with multi-strategy based on energy conversion and explosive mutation, Applied Soft Computing 113 (2021) 107937.
[8] S. Pant, A. Kumar, A. Kishor, D. Anand, S.B. Singh, Application of a multi-objective particle article swarm optimization technique to solve reliability optimization problem, in: in: 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 1004–1007, https://doi.org/10.1109/NGCT.2015.7375272.
[9] A. Kumar, S. Pant, M. Ram, S. Chaube, Multi-objective grey wolf optimizer approach to the reliability-cost optimization of life support system in space capsule, International Journal of System Assurance Engineering and management 51 (2021) 276–284.
[10] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation 2 (1994) 221–248.
[11] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[12] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, technical report (2001).
[13] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (2007) 712–731.
[14] C. Dai, Y. Wang, M. Ye, A new multi-objective particle swarm optimization algorithm based on decomposition, Information Sciences 325 (2015) 541–557.
[15] R. Martín-Moreno, M.A. Vega-Rodríguez, Multi-objective artificial bee colony algorithm applied to the bi-objective orienteering problem, Knowledge-Based Systems 154 (2018) 93–101.
[16] H. Han, W. Lu, J. Qiao, An adaptive multiobjective particle swarm optimization based on multiple adaptive methods, IEEE Transactions on Cybernetics 47 (2017) 2754–2767.
[17] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: Advances in Swarm Intelligence. ICSI 2010. Lecture Notes in Computer Science, volume 6145, 2010, pp. 355–364. DOI: 10.1007/978-3-642-13495-1_44..
[18] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Gsa: A gravitational search algorithm, Information Sciences 179 (2009) 2232–2248. Special Section on High Order Fuzzy Sets..
[19] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Computer Methods in Applied Mechanics and Engineering 376 (2021) 113609.
[20] Q. Yang, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, J. Zhang, A distributed swarm optimizer with adaptive communication for large-scale optimization, IEEE Transactions on Cybernetics 50 (2020) 3393–3408.
[21] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transactions on Cybernetics 45 (2015) 191–204.
[22] S. Pichai, K. Sunat, S. Chiewchanwattana, An asymmetric chaotic competitive swarm optimization algorithm for feature selection in high-dimensional data, Symmetry 12 (2020).
[23] Y. Wang, Z. Yang, Y. Guo, B. Zhou, X. Zhu, A novel binary competitive swarm optimizer for power system unit commitment, Applied Sciences 9 (2019).
[24] Y. Wang, Z. Yang, M. Mourshed, Y. Guo, Q. Niu, X. Zhu, Demand side management of plug-in electric vehicles and coordinated unit commitment: A novel parallel competitive swarm optimization method, Energy Conversion and Management 196 (2019) 935–949.
[25] M.R. Nayak, D. Behura, S. Nayak, Performance analysis of unbalanced radial feeder for integrating energy storage system with wind generator using inherited competitive swarm optimization algorithm, Journal of Energy Storage 38 (2021) 102574.
[26] W.-J. Yin, Z.-F. Ming, Electric vehicle charging and discharging scheduling strategy based on local search and competitive learning particle swarm optimization algorithm, Journal of Energy Storage 42 (2021) 102966.
[27] W. Li, Z. Lei, J. Yuan, H. Luo, Q. Xu, Enhancing the competitive swarm optimizer with covariance matrix adaptation for large scale optimization, Applied Intelligence 51 (2021) 4984–5006.
[28] X. Chen, G. Tang, Solving static and dynamic multi-area economic dispatch problems using an improved competitive swarm optimization algorithm, Energy 238 (2022) 122035.
[29] G. Xiong, J. Zhang, D. Shi, X. Yuan, A simplified competitive swarm optimizer for parameter identification of solid oxide fuel cells, Energy Conversion and Management 203 (2020) 112204.
[30] X. Zhang, X. Zheng, R. Cheng, J. Qiu, Y. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, Information Sciences 427 (2018) 63–76.
[31] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Transactions on Cybernetics 50 (2020) 3696–3708.
[32] W. Huang, W. Zhang, Adaptive multi-objective particle swarm optimization using three-stage strategy with decomposition, Soft Computing 25 (2021) 14645–14672.
[33] L. Deng, L. Song, G. Sun, A competitive particle swarm algorithm based on vector angles for multi-objective optimization, IEEE Access 9 (2021) 89741–89756.
[34] F. Xue, T. Dong, S. You, Y. Liu, H. Tang, L. Chen, X. Yang, J. Li, A hybrid many-objective competitive swarm optimization algorithm for large-scale multirobot task allocation problem, International Journal of Machine Learning and Cybernetics 12 (2021) 943–957.
[35] D. Rajani, V. Kumar, Kumar, Impact of controlling parameters on the performance of mopso algorithm, Procedia Computer Science 167 (2020) 2132–2139.
[36] T.V. Sibalija, Particle swarm optimisation in designing parameters of manufacturing processes: A review (2008–2018), Applied Soft Computing 84 (2019) 105743.
[37] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (2002) 58–73.
[38] L. Li, W. Wang, W. Li, X. Xu, Y. Zhao, A novel ranking-based optimal guides selection strategy in mopso, Procedia Computer Science 91 (2016) 1001–1010.
[39] M.R. Bonyadi, A theoretical guideline for designing an effective adaptive particle swarm, IEEE Transactions on Evolutionary Computation 24 (2020) 57–68.
[40] W. Hu, G.G. Yen, Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system, IEEE Transactions on Evolutionary Computation 19 (2015) 1–18.
[41] Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2009) 1362–1381.
[42] M. Hu, T. Wu, J.D. Weir, An adaptive particle swarm optimization with multiple adaptive methods, IEEE Transactions on Evolutionary Computation 17 (2013) 705–720.
[43] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), volume 1, 2002, pp. 825–830 vol 1. DOI: 10.1109/CEC.2002.1007032..
[44] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (2006) 477–506.
[45] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, S. Tiwari, Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition, technical report, University of Essex, UK, School of Computer Science and Electronic Engineering, 2008.
[46] C.R. Raquel, P.C.N. Jr, An effective use of crowding distance in multiobjective particle swarm optimization, in: Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, June 25–29, 2005, 2005, p. 257–264..

[47] J. Qiao, F. Li, S. Yang, C. Yang, W. Li, K. Gu, An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection, Information Sciences 512 (2020) 446–470.
[48] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (2011) 45–76.
[49] K. Li, R. Chen, G. Fu, X. Yao, Two-archive evolutionary algorithm for constrained multiobjective optimization, IEEE Transactions on Evolutionary Computation 23 (2019) 303–315.