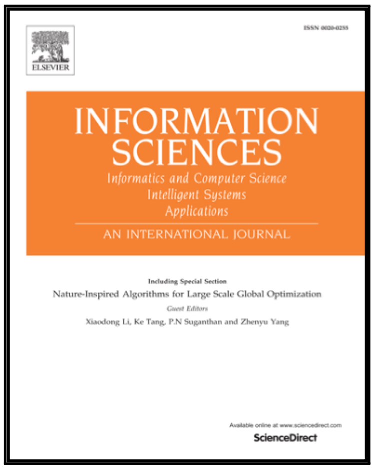
A competitive mechanism based multi-objective particle swarm optimizer with fast convergence

Zhang, Xingyi; Zheng, Xiutao; Cheng, Ran; Qiu, Jianfeng; Jin, Yaochu

DOI:
10.1016/j.ins.2017.10.037



A Competitive Mechanism Based Multi-objective Particle Swarm Optimizer with Fast Convergence

Xingyi Zhang, Xiutao Zheng, Ran Cheng, Jianfeng Qiu, Yaochu Jin

PII: S0020-0255(17)31034-4
DOI: 10.1016/j.ins.2017.10.037
Reference: INS 13212

To appear in: *Information Sciences*

Received date: 26 June 2017
Revised date: 14 September 2017
Accepted date: 15 October 2017

Please cite this article as: Xingyi Zhang, Xiutao Zheng, Ran Cheng, Jianfeng Qiu, Yaochu Jin, A Competitive Mechanism Based Multi-objective Particle Swarm Optimizer with Fast Convergence, *Information Sciences* (2017), doi: 10.1016/j.ins.2017.10.037

A Competitive Mechanism Based Multi-objective Particle Swarm Optimizer with Fast Convergence

Xingyi Zhang[a], Xiutao Zheng[a], Ran Cheng[b,*], Jianfeng Qiu[a], Yaochu Jin[c]

[a]*Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China*
[b]*School of Computer Science, University of Birmingham, Birmingham, B15 2TT, U.K.*
[c]*Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, U.K.*

Abstract

In the past two decades, multi-objective optimization has attracted increasing interests in the evolutionary computation community, and a variety of multi-objective optimization algorithms have been proposed on the basis of different population based meta-heuristics, where the family of multi-objective particle swarm optimization is among the most representative ones. While the performance of most existing multi-objective particle swarm optimization algorithms largely depends on the global or personal best particles stored in an external archive, in this paper, we propose a competitive mechanism based multi-objective particle swarm optimizer, where the particles are updated on the basis of the pairwise competitions performed in the current swarm at each generation. The performance of the proposed competitive multi-objective particle swarm optimizer is verified by benchmark comparisons with several state-of-the-art multi-objective optimizers, including three multi-objective particle swarm optimization algorithms and three multi-objective evolutionary algorithms. Experimental results demonstrate the promising performance of the proposed algorithm in terms of both optimization quality and convergence speed.

Keywords: Multi-objective optimization, Competitive swarm optimizer,

*Corresponding author.
Email addresses: xyzhanghust@gmail.com (Xingyi Zhang), zxt0086@outlook.com (Xiutao Zheng), ranchengcn@gmail.com (Ran Cheng), qiujianf@ahu.edu.cn (Jianfeng Qiu), yaochu.jin@surrey.ac.uk (Yaochu Jin)

Evolutionary algorithm, Particle swarm optimization

1. Introduction

In the real world, many optimization problems may involve multiple conflicting objectives to be optimized simultaneously [10, 11, 14, 29, 30, 46, 49]. Such optimization problems are usually called multi-objective optimization problems (MOPs), which are generally more challenging to be solved than single-objective optimization problems (SOPs), since there usually exist a set of solutions to be obtained as trade-offs between different objectives for MOPs [39].

In the past two decades, multi-objective optimization has attracted increasing interests in the evolutionary computation community, and a large number of multi-objective optimization algorithms have been developed on the basis of different population based meta-heuristics, such as genetic algorithm [41], immune clone algorithm [31], differential evolution algorithm [1], firefly algorithm [17] and neural network regression [9]. It is worth noting that nature-inspired optimization algorithms have also been extensively applied to solve other optimization problems, e.g., creation of graphic characters [18], optimal outcome of evolutionary games [45] and inventory control [34, 35, 36, 38]. Particle swarm optimization (PSO) [20], as one of the most classical swarm intelligence algorithms, has been widely applied to solve SOPs due to its simple implementation and fast convergence. Moreover, as reported in some recent studies [5, 6, 25], PSO also has good potential in solving MOPs.

In order to apply PSO to multi-objective optimization, there are at least two fundamental issues to be addressed. The first issue is how to define the personal and global best particles, given that there does not exist any particle which can perform the best on all objectives of an MOP. Since the personal and global best particles are used to guide the search direction of particles in the swarm, they have considerable influence on the performance of PSO algorithms, especially in solving MOPs [6]. The second issue is how to balance convergence and diversity of the swarm. Since the target of multi-objective optimization is to obtain a set of trade-off solutions, diversity maintenance is particularly important. A PSO based multi-objective algorithm is very likely to be trapped into local optimum (or one of the many optima) of an MOP due to its fast convergence. Therefore, striking a balance between convergence and diversity is curial to the performance of multi-objective PSO algorithms [23].

In the past ten years, a lot of multi-objective PSO algorithms have been suggested by addressing the above two issues [24, 25, 28, 32, 43], which can be roughly divided into two categories. The first category is to define the personal and global best particles based on the Pareto ranking scheme [19]. Three representatives of this category are multi-objective particle swarm optimization [5], improved multi-objective particle swarm optimizer [32] and speed-constrained multi-objective PSO [25]. There are also some multi-objective PSO algorithms proposed on the basis of some enhanced ranking schemes, such as global margin ranking [22] and preference order ranking [42]. In these algorithms, an archive is maintained to store elite particles determined by the ranking schemes and these elite particles are used as candidates for personal and global best particles. The second category adopts the decomposition strategy to transform MOPs into a set of SOPs, such that the single-objective PSO algorithms can be directly applied to multi-objective optimization. The first decomposition based multi-objective PSO algorithm was suggested by Parsopoulos and Vrahatis based on dynamic weighted aggregation [16, 27]. Recently, several improved multi-objective PSO algorithms based on decomposition were also reported in the literature [6, 23, 24, 28]. Generally, the multi-objective PSO algorithms as mentioned above can achieve a good balance between convergence and diversity for most MOPs, but still encounter great challenges when tackling complex MOPs, especially for those with a large number of local optima (e.g., DTLZ1 and DTLZ3 [8]).

To further enhance the robustness of PSO in solving MOPs, in this paper we suggest a multi-objective PSO algorithm inspired by the recently developed competitive swarm optimizer [2]. The competitive swarm optimizer is a variant of PSO and the main difference lies in the fact that the search process is guided by the competitors in the current swarm instead of the historical positions, i.e., the personal and global best particles. Both theoretical analysis and empirical results have demonstrated that the competitive swarm optimizer is able to achieve a better balance between convergence and diversity than original PSO by adopting the competition mechanism [2]. By taking advantage of such a competition mechanism, in this paper, we propose a competitive mechanism based multi-objective PSO, termed CMOPSO, where a competition mechanism based learning strategy is designed to guide the search of PSO for multi-objective optimization. In summary, the main contributions of this paper are as follows.

(1) A competition mechanism based learning strategy is suggested for the updating of particles in multi-objective PSO. In the proposed strategy, pairwise competitions are randomly performed between particles in the current swarm. The winner particle is used to guide the particle by updating the velocity accordingly. Compared to the updating strategies in existing multi-objective PSO algorithms, the proposed competition mechanism based learning strategy achieves better balance between convergence and diversity.
(2) A novel multi-objective PSO algorithm, called CMOPSO, is proposed on the basis of the competition mechanism based learning strategy. In CMOPSO, no additional storage is required to record the historical information in the search process, such that it does not need any external archive. By contrast, most existing multi-objective PSO algorithms often need to maintain an archive with a high computational cost, e.g., multi-objective PSO algorithms developed in [23, 25, 32, 43].
(3) The performance of the proposed CMOPSO is verified by comparing it with six existing algorithms on 21 benchmark MOPs, including three popular multi-objective PSO algorithms, namely, MPSOD [6], MMOPSO [23], SMPSO [25] and three well-known multi-objective evolutionary algorithms (MOEAs), namely, NSGA-II [7], MOEA/D [47] and SPEA2 [53]. The experimental results demonstrate that the proposed CMOPSO shows significantly better overall performance than the compared algorithms, in terms of both quality of solution set and convergence speed.

The rest of this paper is organized as follows. In Section 2, we review a few multi-objective PSO algorithms and briefly introduce the competitive swarm optimizer. The details of the proposed CMOPSO are given in Section 3 and the performance of CMOPSO is verified in Section 4 by comparing it with existing multi-objective PSO algorithms and MOEAs. Finally, conclusion and future work are presented in Section 5.

2. Related work

2.1. Existing multi-objective PSO algorithms

PSO is a well-known swarm intelligence paradigm originally inspired by the behavior of bird flocking in nature, and later has been widely applied to solve SOPs [15, 26, 33, 37, 44]. Due to the high convergence speed and simple implementation, recently, a number of multi-objective PSO algorithms have also been proposed in the literature [5, 6, 23, 24, 25, 28, 32, 43]. In the following, we briefly review some representative multi-objective PSO algorithms.

The first PSO based multi-objective algorithm was suggested by Coello Coello *et al.* in [5]. In the algorithm, the concept of Pareto dominance was suggested to determine the global and personal best particles, and an archive was maintained to save the non-dominated particles as global best particles. Although the multi-objective PSO algorithm has demonstrated competitive performance in solving MOPs in comparison with traditional MOEAs such as NSGA-II [7] and PAES [21], it has difficulties in solving MOPs with complicated landscapes, e.g., those with multiple local fronts. To address this issue, Sierra and Coello Coello [32] proposed an improved PSO based multi-objective algorithm, where different mutation operators were suggested for different subswarms divided by users in advance. Experimental results showed that the improved algorithm performs better than the first multi-objective PSO algorithm on MOPs with multiple local fronts.

Nebro *et al.* [25] developed a speed-constrained multi-objective PSO algorithm, called SMPSO, where the velocities of all particles were limited in order to tackle MOPs with multimodal landscapes. As reported in [25], most multi-objective PSO algorithms often suffer from an issue called “swarm explosion” due to the fact that the velocities are too high such that the particles tend to move towards the boundaries of the search space. Therefore, the speed constraint is an effective strategy to enhance the performance of PSO based multi-objective algorithms.

In contrast to the multi-objective PSO algorithms where the global and personal best particles are determined by dominance relations, Peng and Zhang proposed a multi-objective PSO algorithm by decomposing an MOP into a number of SOPs [28]. The decomposition based multi-objective PSO algorithm followed the framework of MOEA/D [47] and replaced the genetic operators with the PSO based search approach. In the algorithm, an external archive was maintained to store the global best particle of each SOP. Motivated by the decomposition mechanism, Martinez and Coello Coello [24] also proposed a multi-objective particle swarm optimizer, where a re-initialization strategy was suggested to maintain the swarm diversity. Since the global best particles in the algorithm were selected from the current swarm, it holds a lower computational cost than most other multi-objective PSO algorithms which often need to maintain an archive. However, as pointed out in [24], the multi-objective PSO algorithm suggested in [24] may fail to find solutions covering the entire Pareto fronts for some complex MOPs due to the lack of an archive. As another decomposition based multi-objective PSO algorithm, Dai *et al.* [6] proposed to divide the objective space into a set of sub-regions based on a set of direction vectors, and at most one particle is maintained in each sub-region.

There are also some other approaches proposed for enhancing swarm diversity in multi-objective PSO algorithms. Zhan *et al.* [43] developed a multi-objective PSO algorithm based on a coevolutionary technique. Different from most existing multi-objective PSO algorithms where multiple objectives were considered as a whole, the algorithm in [43] provided a simple and straightforward way to solve MOPs by letting each population correspond with only one objective. In the algorithm, an external shared archive was adopted to exchange information in different populations for enhancing the diversity to avoid local optima. Wang and Yen [12] developed an adaptive multi-objective PSO algorithm, where the balance between convergence and diversity was achieved by dynamically adjusting the exploration and exploitation according to the feedback information detected from the evolutionary environment by a parallel cell coordinate system. Lin and Li [23] proposed a multi-objective PSO algorithm with multiple search strategies, where the PSO based search approach was performed on the particles in the swarm, and the genetic operators, namely, simulated binary crossover [7] and polynomial mutation [54], were adopted to update the particles in the external archive.

As reviewed above, one of the main concerns in existing multi-objective PSO algorithms is how to effectively enhance swarm diversity for tackling MOPs with local fronts or multimodal landscapes. In this paper, we propose a competitive mechanism based multi-objective PSO algorithm (termed CMOPSO), which is inspired from the recently proposed competitive swarm optimizer [2]. In the proposed CMOPSO, the particles are updated on the basis of a competition mechanism based learning strategy instead of the global and personal best particles, and thus there is no external archive maintained. The details of CMOPSO will be presented in Section 3.

2.2. Competitive swarm optimizer

The competitive swarm optimizer is a variant of PSO proposed by Cheng and Jin in [2] for dealing with SOPs. In the competitive swarm optimizer, the particles are updated via a competition mechanism instead of using the global and personal best particles, thus substantially improving the swarm diversity to avoid premature convergence. To be specific, in the competitive swarm optimizer, particles are pairwise randomly selected from the current swarm for competition and the loser in the competition is updated by learning from the winner, while the winner is directly passed to the swarm of next generation.

It is assumed that there are N particles in the swarm $P(t)$, where t is the generation index. Each particle has an n-dimensional position, $X_i(t) = (x_{i,1}(t), x_{i,2}(t), \ldots, x_{i,n}(t))$ and n-dimensional velocity vector, $V_i(t) = (v_{i,1}(t), v_{i,2}(t), \ldots, v_{i,n}(t))$. In the k-th round of the competition in generation t, the positions and the velocities of the winner and loser are denoted as $X_{w,k}(t)$, $X_{l,k}(t)$, $V_{w,k}(t)$ and $V_{l,k}(t)$, respectively, where $k = 1, 2, \ldots, N/2$. Accordingly, after the k-th competition the loser's velocity will be updated using the following learning strategy:

$$\begin{aligned} V_{l,k}(t+1) = {} & R_1(k,t)V_{l,k} \\ & + R_2(k,t)(X_{w,k}(t) - X_{l,k}(t)) \\ & + \varphi R_3(k,t)(\overline{X}_k(t) - X_{l,k}(t)) \end{aligned} \tag{1}$$

As a result, the position of the loser can be updated on the basis of the new velocity:

$$X_{l,k}(t+1) = X_{l,k}(t) + V_{l,k}(t+1) \tag{2}$$

where $R_1(k,t), R_2(k,t), R_3(k,t)$ are random numbers generated uniformly in the range [0,1], $\overline{X}_k(t)$ is the mean position of all particles in the swarm and φ is a parameter for controlling the influence of $\overline{X}_k(t)$.

Algorithm 1: General framework of CMOPSO

Input: N (swarm size)
Output: P (final positions)

```
1: P ← RandomInitialize(N);
2: V ← RandomInitialize(N);
3: while termination criterion not fulfilled do
4:    P′ ← CompetitionBasedLearning(P, V);
5:    P ← EnvironmentalSelection(P, P′);
6: end while
7: return P
```

3. The proposed CMOPSO

In this section, we first present the framework of the proposed CMOPSO, and then elaborate the details of the competition mechanism based learning strategy suggested in CMOPSO for multi-objective PSO algorithms.

3.1. The framework of CMOPSO

As presented in Algorithm 1, the proposed CMOPSO has a very simple framework, where the main loop consists of two main components: the competition mechanism based learning strategy and the environmental selection. For simplicity, we directly adopt the environmental selection as suggested in SPEA2 [53], while the details of the competition mechanism based learning strategy are presented as follows.

3.2. The competition mechanism based learning strategy

The proposed competition mechanism based learning strategy consists of three components, namely, elite particle selection, pairwise competition and particle learning. Algorithm 2 presents the pseudo code of the competition mechanism based learning strategy. In what follows, we give the details of each component respectively.

Since the elite particle set is used to provide candidate particles to be used in the pairwise competitions to guide the search of the swarm, the particles in it should maintain good balance between convergence and diversity. For simplicity, in this paper, the elite particles are selected via the non-dominated sorting and crowding distance based ranking as adopted

Algorithm 2: $CompetitionBasedLearning(P, V)$

Input: P (current positions), V (current velocities);
L (elite particle set), γ (size of elite particle set);
Output: P' (new positions)

$P' \leftarrow \emptyset$;
/*Elite Particles Selection*/
$L \leftarrow$ Select γ particles from P according to the front index and the crowding distance of each particle;
for each particle $p_i \in P$ **do**
  randomly choose two elite particles a, b from L;
  /*Pairwise Competition*/
  calculate the angle θ_1 between a and p_i, and θ_2 between b and p_i;
  if $\theta_1 < \theta_2$ **then**
    $p_w \leftarrow a$;
  else
    $p_w \leftarrow b$;
  end if
  /*Particle Learning*/
  $v_i' \leftarrow$ update the velocity of particle p_i using formula 3;
  $p_i' \leftarrow$ update the position of particle p_i using formula 4;
  $P' \leftarrow P' \cup \{p'\}$;
end for
/*Mutation*/
$P' \leftarrow PolynomialMutation$ (P');
return P'

in NSGA-II [7]. Specifically, the non-dominated sorting [50] is first performed on the swarm P to obtain the fronts $F_1, F_2, \ldots, F_k$, where k is the maximum index of fronts. Then, the minimum number t is found such that $|F_1 \cup F_2 \cup \ldots \cup F_t| \geq \gamma$, where γ is the number of elite particles to be selected. Finally, all particles belonging to $F_1 \cup F_2 \cup \ldots \cup F_{t-1}$ are selected as the elite particles and the remaining particles are selected from F_t according to the crowding distance of each particle in F_t.

It is worth noting that, since the elite particles are selected from the current swarm at each generation, the proposed CMOPSO does not need any additional external archive, whereas most existing multi-objective PSO al-

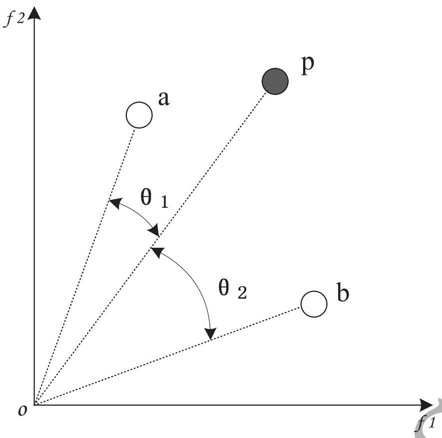


Figure 1: An example illustrating the pairwise competition between two randomly selected elite particles. In this example, p is a particle to be updated, a and b are two randomly selected competitors from elite particle set. θ_1 is the angle between p and a, and θ_2 is the angle between p and b. Consequently, particle a wins the competition since $\theta_1 < \theta_2$.

gorithms often contain an archive to store the global and personal best partilces. In addition, the elite particle set size γ can be used to control the convergence speed of the proposed CMOPSO. For MOPs, a small value of γ can lead to the premature convergence, whereas a large value will retard the convergence speed of the algorithm. A detailed discussion on γ will be given in Section 4.

After the elite particle set is created, pairwise competitions are performed among the particles in it, and the winners will be used to guide the moving directions of particles in the current swarm. For each pairwise competition, given a particle p in the swarm, two elite particles a and b are randomly selected from the elite particle set. The angles between a, b and p are calculated respectively, and the elite particle with a smaller angle wins the competition, such that the particle p will learn from the elite particle which is closer to the convergence direction of it. Figure 1 presents an illustrative example of the pairwise competition in the proposed CMOPSO, where a and b are two randomly selected competitors from the elite particle set and p is a particle in the swarm to be updated. As shown in the figure, the competitor a will become the winner since angle θ_1 between a and p is smaller than angle θ_2 between b and p, and thus a will be used to update p.

Once the winner is determined, the position and velocity of particle p can be updated via learning the winner. Let v_i and p_i be the velocity and position of the i-th particle in the swarm, $1 \leq i \leq N$. The updated velocity v_i' and position p_i' of the i-th particle are calculated using the following equations as suggested in competitive swarm optimizer [2]:

$$v_i' = R_1 v_i + R_2(p_w - p_i), \tag{3}$$

$$p_i' = p_i + v_i', \tag{4}$$

where $R_1, R_2 \in [0, 1]$ are two randomly generated vectors and p_w is the position of the winner. Finally, similar to most existing multi-objective PSO algorithms [23, 25], the polynomial mutation [54] is also performed.

4. Experimental studies

In this section, we verify the performance of CMOPSO by comparing it with three existing multi-objective PSO algorithms, MPSOD [6], MMOPSO [23] and SMPSO [25], and three popular MOEAs, NSGA-II [7], MOEA/D [47] and SPEA2 [53]. A total of 21 benchmark MOPs from three test suits, ZDT [52], DTLZ [8] and WFG [13] are used to evaluate the performance of the algorithms, where ZDT1 to ZDT4 and ZDT6 are bi-objective problems and two-/three-objective DTLZ and WFG test problems are considered. For bi-objective problems, the number of decision variables is set to 30 in ZDT1 to ZDT3, to 10 in ZDT4 and ZDT6, to 6 in DTLZ1, to 21 in DTLZ7, to 11 in DTLZ2 to DTLZ6 and all WFG problems. For three-objective problems, the number of decision variables is fixed to 12 in DTLZ2 to DTLZ6 and all WFG test problems, to 7 in DTLZ1 and to 22 in DTLZ7.

The inverted generational distance (IGD) metric [48, 51] is adopted to evaluate the performance of the compared algorithms, and roughly 5000 points uniformly sampled on the Pareto fronts are used in the calculation of IGD for each test problem. The IGD is a metric for evaluating the quality of obtained solution set in terms of both convergence and distribution. The smaller the IGD value, the better the quality of solution set obtained by an algorithm.

For fair comparisons, all parameters of the compared algorithms are set to the recommended values as in their original papers. The population size is set to $N = 100$ for all compared algorithms. The parameter γ in the proposed CMOPSO is set to 10 for each test problem in the experiments. The number of generations is adopted as the termination criterion for all considered algorithms. For DTLZ3, the maximal number of generations is set to 1,000, and to 300 for the rest test problems. On each test instance, 30 independent runs are conducted and the median value is reported. All the experimental results are obtained on a PC with an Intel Core i5 4590 3.30GHz CPU and Microsoft Windows 7 operating system. The source codes of NSGA-II, SPEA2 and MOEA/D are provided in PlatEMO [40], and the source codes of MPSOD, SMPSO and MMOPSO are obtained from the authors of the original papers.

4.1. Comparisons with existing multi-objective PSO algorithms

Table 1 presents the mean and standard deviation of IGD values of the four PSO based multi-objective algorithms on ZDT1 to ZDT4, ZDT6, DTLZ1 to DTLZ7 and WFG1 to WFG9. Moreover, the Wilconxon rank sum test is adopted at a significance level of 0.05, where the symbols '+', '-' and '$\approx$' indicate that the result is significantly better, significantly worse and statistically similar to that obtained by CMOPSO, respectively.

It can be observed that the proposed CMOPSO performs better than the three compared multi-objective PSO algorithms, SMPSO, MPSOD and MMOPSO on the benchmark test problems. On all 37 test instances under consideration, the CMOPSO achieved statistically significantly better IGD value on 18 test instances, MPSOD on 6 test instances, MMOPSO on 8 test instances and SMPSO on 1 test instance. There are also 4 test instances on which both the proposed CMOPSO and MPSOD obtained significantly better IGD value than the other two compared algorithms. It can also be seen that the proposed CMOPSO is suited to solving test problems with multiple local fronts. On DTLZ1 and DTLZ3 with two objectives, the proposed CMOPSO can obtain a set of non-dominated solutions which can well approximate the whole Pareto front and maintain with a good distribution. This can also be observed from Figure 2, which plots the non-dominated solution sets associated with the best IGD value among 30 runs for the CMOPSO and the three compared multi-objective PSO algorithms on 2-objective DTLZ3 in objective space.

For three-objective DTLZ1, the proposed CMOPSO can still achieve a competitive performance, but it seems that the performance of CMOPSO has considerably deteriorated on three-objective DTLZ3. The main reason is attributed to the fact that the three-objective DTLZ3 considered here

Table 1: IGD values of the proposed CMOPSO and three multi-objective PSO algorithms on ZDT, DTLZ and WFG test problems, where the best mean value on each test instance is highlighted in a gray background. Each column presents the IGD values of an algorithm on different problems. The experimental results in the table demonstrate the superiority of the proposed CMOPSO over the three compared multi-objective PSO algorithms.

Problem	Obj.	SMPSO	MMOPSO	MPSOD	CMOPSO
ZDT1	2	4.84e-3(1.26e-4)−	4.54e-3(9.03e-5)−	3.82e-3(3.96e-6)≈	3.82e-3(2.15e-5)
ZDT2		4.79e-3(1.10e-4)−	2.91e-2(1.07e-1)−	3.82e-3(2.18e-5)+	3.86e-3(1.62e-5)
ZDT3		5.09e-3(1.11e-4)−	5.20e-3(1.36e-4)−	9.90e-3(4.22e-5)−	4.50e-3(2.83e-5)
ZDT4		1.25e-0(5.81e-1)−	8.09e-3(2.98e-3)+	1.87e+1(3.12e-0)−	3.70e-2(4.59e-2)
ZDT6		3.67e-3(1.03e-4)−	3.62e-3(6.53e-5)−	3.09e-3(8.53e-7)≈	3.09e-3(2.61e-5)
DTLZ1	2	2.04e-0(2.36e-0)−	2.22e-3(8.94e-5)+	1.06e-0(3.14e-1)−	4.44e-2(7.83e-2)
DTLZ2		5.12e-3(1.97e-4)−	4.92e-3(1.76e-4)−	4.15e-3(3.08e-5)+	4.40e-3(3.61e-5)
DTLZ3		1.67e+1(2.29e+1)−	4.97e-3(1.81e-4)−	5.02e-0(1.46e-0)−	4.24e-3(1.71e-4)
DTLZ4		3.49e-1(3.74e-1)−	4.92e-3(1.67e-4)−	4.12e-3(6.04e-5)+	4.41e-3(7.58e-5)
DTLZ5		5.07e-3(1.78e-4)−	5.03e-3(1.62e-4)−	4.13e-2(3.58e-5)−	4.40e-3(5.66e-5)
DTLZ6		5.16e-3(2.63e-4)−	5.01e-3(1.39e-4)+	3.96e-3(1.74e-6)+	4.09e-3(1.83e-5)
DTLZ7		1.95e-1(2.21e-1)−	3.43e-2(1.11e-1)−	6.57e-2(1.64e-5)−	4.43e-3(4.01e-5)
WFG1	2	1.12e+0(3.83e-2)−	3.26e-1(4.10e-2)+	1.35e-0(4.81e-2)−	9.62e-1(6.74e-2)
WFG2		1.89e-2(3.44e-3)−	1.17e-2(2.67e-4)−	4.32e-2(4.14e-3)−	1.12e-2(2.33e-4)
WFG3		1.75e-2(6.78e-4)−	1.43e-2(4.84e-4)−	2.07e-2(2.33e-3)−	1.32e-2(2.49e-4)
WFG4		4.95e-2(4.72e-4)−	1.70e-2(5.47e-4)+	4.09e-2(1.37e-3)−	3.88e-2(7.95e-3)
WFG5		6.67e-2(7.25e-4)−	6.49e-2(3.37e-3)−	6.54e-2(1.64e-3)−	6.46e-2(3.41e-4)
WFG6		2.02e-2(3.10e-3)−	2.57e-2(6.65e-3)−	2.53e-2(3.58e-3)−	1.72e-2(3.01e-3)
WFG7		1.63e-2(4.12e-4)−	1.59e-2(6.68e-4)−	1.88e-2(8.91e-4)−	1.38e-2(2.38e-4)
WFG8		1.01e-1(2.16e-3)+	1.09e-1(1.26e-3)+	1.13e-1(2.79e-3)+	1.16e-1(1.78e-3)
WFG9		2.61e-2(1.38e-3)−	2.21e-2(1.03e-3)+	2.67e-2(1.46e-3)−	2.50e-2(2.07e-3)
DTLZ1	3	3.41e-0(3.28e-0)−	7.62e-2(9.44e-2)−	9.68e-1(1.38e-1)−	4.43e-2(7.83e-2)
DTLZ2		5.12e-2(2.13e-4)−	6.89e-2(1.18e-3)−	5.47e-2(8.48e-5)−	4.33e-3(4.34e-5)
DTLZ3		8.73e-0(1.42e+1)−	6.87e-2(1.17e-3)+	8.01e-0(2.86e-0)−	1.23e-0(9.68e-1)
DTLZ4		2.99e-1(3.67e-1)−	6.88e-2(1.20e-3)−	5.47e-2(1.58e-4)−	4.41e-3(7.58e-5)
DTLZ5		5.12e-3(1.78e-4)−	5.90e-3(1.57e-4)−	3.43e-2(1.80e-3)−	4.32e-3(3.97e-5)
DTLZ6		5.13e-3(2.97e-4)−	5.73e-3(2.66e-4)−	3.11e-2(7.71e-3)−	4.16e-3(2.49e-5)
DTLZ7		1.80e-1(2.18e-1)−	7.93e-2(2.03e-3)−	1.25e-1(2.27e-3)−	4.43e-2(4.43e-5)
WFG1	3	1.52e-0(1.78e-2)−	8.29e-1(6.54e-2)+	1.58e-0(3.82e-2)≈	1.58e-0(2.50e-2)
WFG2		2.11e-1(1.62e-2)−	2.04e-1(8.98e-3)−	3.66e-1(7.14e-2)−	1.56e-1(7.49e-3)
WFG3		1.29e-1(1.57e-2)+	8.52e-2(1.03e-2)+	2.57e-1(8.71e-3)−	1.50e-1(1.04e-2)
WFG4		3.08e-1(5.46e-3)−	2.97e-1(5.59e-3)−	2.59e-1(1.29e-3)≈	2.59e-1(3.61e-3)
WFG5		2.84e-1(7.94e-3)−	2.82e-1(5.07e-3)−	2.53e-2(4.88e-4)+	2.35e-1(2.49e-3)
WFG6		3.04e-1(1.51e-2)−	2.98e-1(1.38e-2)−	2.64e-2(4.01e-3)+	2.41e-1(7.72e-3)
WFG7		2.96e-1(6.93e-3)−	2.79e-1(5.89e-3)−	2.56e-1(1.94e-3)−	2.26e-1(3.20e-3)
WFG8		4.51e-1(1.60e-2)−	3.64e-1(4.71e-3)−	3.32e-1(4.18e-3)≈	3.32e-1(5.33e-3)
WFG9		2.69e-1(6.06e-3)−	2.77e-1(6.55e-2)−	2.45e-1(7.47e-4)−	2.17e-1(2.24e-3)
+/ − / ≈		2/35/0	10/27/0	7/25/5	

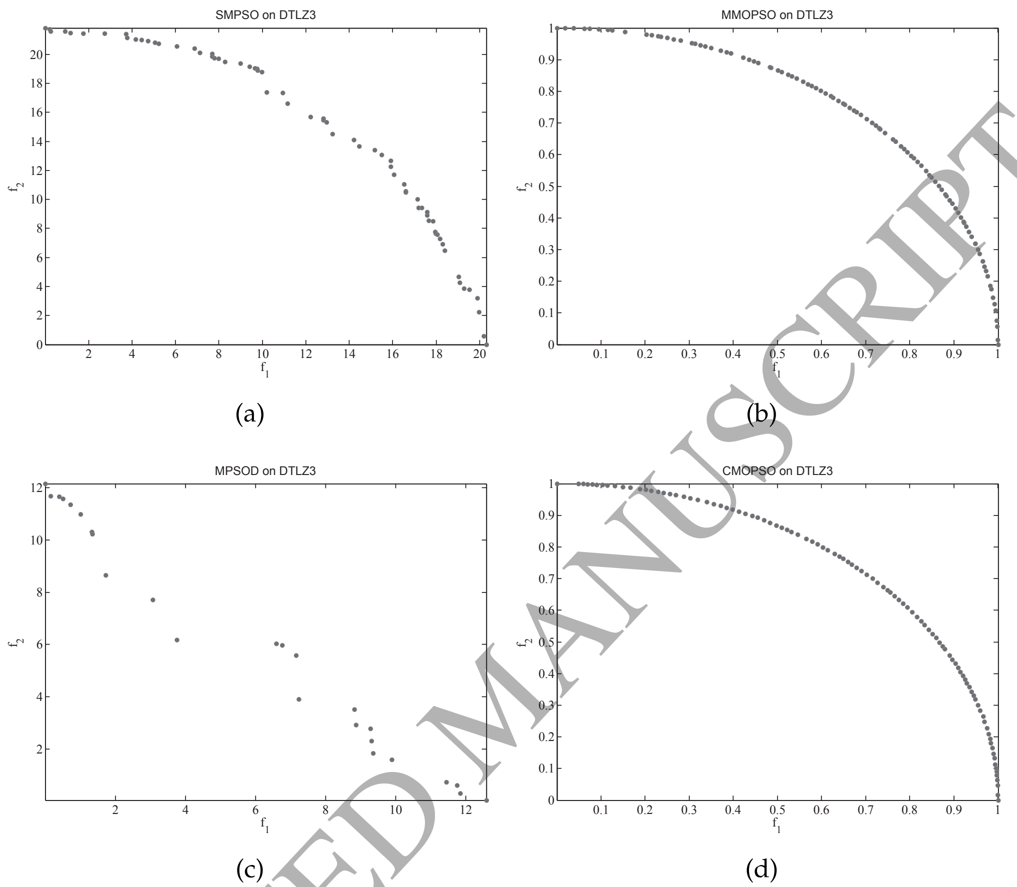
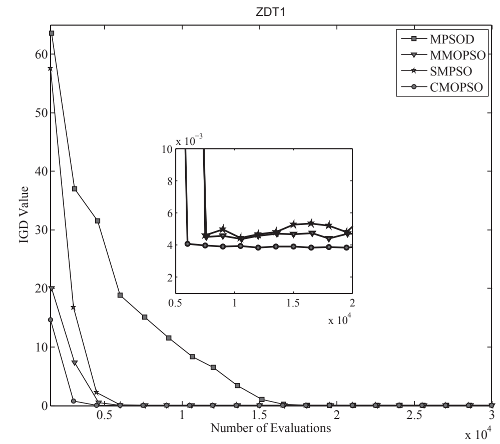
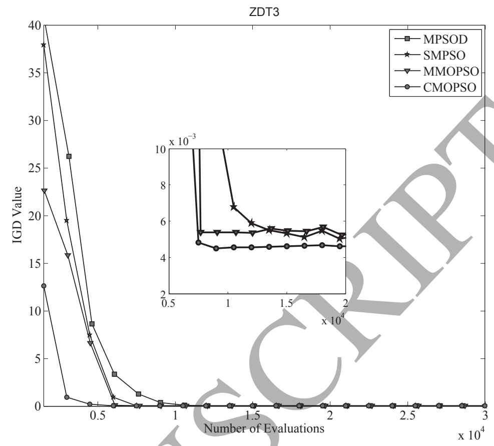


Figure 2: The non-dominated solution sets associated with the best IGD value among 30 runs obtained by the proposed CMOPSO and three compared multi-objective PSO algorithms on bi-objective DTLZ3. (a) The non-dominated solution sets obtained by SMPSO; (b) The non-dominated solution sets obtained by MMOPSO; (c) The non-dominated solution sets obtained by MPSOD; (d) The non-dominated solution sets obtained by CMOPSO. The experimental results show that the proposed CMOPSO outperforms the three compared multi-objective PSO algorithms in terms of both convergence and diversity on bi-objective DTLZ3.

is much more challenging than the three-objective DTLZ1 due to a large number of $11^{10} - 1$ local fronts, whereas the DTLZ1 contains a number of $11^5 - 1$ local fronts. It is worth noting that the MMOPSO performs the best on DTLZ1 and DTLZ3 due to the fact that it has adopted the crossover and mutation operators in MOEAs in addition to the updating strategies



(a)



(b)

Figure 3: Convergence trajectories of the four compared multi-objective PSO algorithms on ZDT1 and ZDT3, averaging over 30 runs. (a) Convergence trajectory of the four compared multi-objective PSO algorithms on ZDT1; (b) Convergence trajectory of the four compared multi-objective PSO algorithms on ZDT3. The experimental results indicate the promising convergence speed of the proposed CMOPSO in comparison with the three multi-objective PSO algorithms on ZDT1 and ZDT3.

of PSO. Generally speaking, compared with existing multi-objective PSO algorithms, the proposed CMOPSO demonstrates the overall best performance, especially on MOPs with local fronts.

In addition to the quality of solution set, another important performance metric of a multi-objective PSO algorithm is its convergence speed. In the following, we verify the convergence speed of the proposed CMOPSO by comparing it with existing multi-objective PSO algorithms. Figure 3 plots the convergence trajectories of the proposed CMOPSO and three compared multi-objective PSO algorithms on ZDT1 and ZDT3, averaging over 30 runs. It can be observed that the proposed CMOPSO has a promising convergence speed, which confirms that the pairwise competition suggested in CMOPSO is able to well balance the convergence and diversity.

From the above empirical results, we can conclude that the proposed CMOPSO is promising in comparison with existing multi-objective PSO algorithms in solving MOPs, especially for solving those with local fronts.

4.2. Comparison with existing MOEAs

Table 2 presents the mean and standard deviation of IGD values of NSGA-II, MOEA/D, SPEA2 and CMOPSO on ZDT1 to ZDT4 and ZDT6, DTLZ1 to DTLZ7 and WFG1 to WFG9, where the Wilconxon rank sum test is also adopted and the best mean for each test instance is highlighted with a gray background. As evidenced by the results, the proposed CMOPSO has also achieved promising overall performance on the benchmark test problems in comparison with existing popular MOEAs, where it has obtained statistically significantly best performance on 18 out of 37 test instances. Besides, the performance of CMOPSO on MOPs with local fronts is also comparable with the compared MOEAs. For bi-objective DTLZ3, CMOPSO performs the best among the four algorithms. As for DTLZ1 with two and three objectives, the CMOPSO also achieves a competitive performance despite that the IGD value is slightly worse. However, the CMOPSO is significantly outperformed by the compared MOEAs on three-objective DTLZ3, which is due to the fact that the genetic operators are more suitable than PSO operator for solving MOPs with local fronts. This is the main reason that some researchers have suggested to adopt genetic operators in existing multi-objective PSO algorithms, such as MMOPSO [23].

In the following, we compare the convergence speed of the proposed CMOPSO and the three MOEAs. Figure 4 plots the convergence trajectories of the four algorithms on ZDT1 and ZDT3, averaging over 30 runs. As shown in the figure, the proposed CMOPSO shows the fastest convergence speed, which enables it to converge to the Pareto fronts of ZDT1 and ZDT3 even after a very small number of generations. As further observations, Figures 5 and 6 present the non-dominated solution sets associated with the best IGD value among 30 runs obtained by the CMOPSO and three MOEAs after 50 generations on ZDT1 and ZDT3, respectively. It can be clearly seen that the proposed CMOPSO can obtain a set of non-dominated solutions evenly distributed on the Pareto fronts of ZDT1 and ZDT3 after 50 generations, whereas the compared MOEAs, namely, NSGA-II, MOEA/D and SPEA2, are still far from convergence.

4.3. Parameter analysis

In the proposed CMOPSO, there is an important parameter γ which controls the size of elite particle set. This parameter has some influence on the performance of the proposed CMOPSO by balancing the convergence and diversity of the swarm. A much small elite particle set may cause

Table 2: IGD values of the proposed CMOPSO and three MOEAs on ZDT, DTLZ and WFG test problems, where the best mean value on each test instance is highlighted in a gray background. Each column presents the IGD values of an algorithm on different problems. The experimental results in the table show the competitiveness of the proposed CMOPSO in comparison with the three popular MOEAs.

Problem	Obj.	NSGA-II	MOEA/D	SPEA2	CMOPSO
ZDT1	2	4.55e-3(1.75e-4)−	8.18e-3(6.02e-3)−	3.88e-3(8.32e-5)−	3.82e-3(2.15e-5)
ZDT2		4.74e-3(2.38e-4)−	9.09e-3(6.93e-3)−	3.89e-3(1.12e-4)−	3.86e-3(1.62e-5)
ZDT3		3.46e-2(4.09e-2)−	1.73e-2(9.19e-3)−	7.83e-3(2.91e-4)−	4.50e-3(2.83e-5)
ZDT4		5.93e-3(1.32e-3)+	2.63e-2(1.83e-3)+	5.07e-3(2.19e-5)+	3.70e-2(4.59e-2)
ZDT6		3.68e-3(1.29e-4)−	6.47e-3(1.22e-3)−	3.15e-3(9.14e-5)−	3.09e-3(2.61e-5)
DTLZ1	2	2.21e-3(1.10e-4)+	2.75e-3(8.28e-4)+	2.24e-3(4.81e-4)+	4.44e-2(7.83e-2)
DTLZ2		5.08e-3(2.06e-4)−	3.99e-3(2.53e-7)+	4.18e-3(4.65e-5)+	4.40e-3(3.61e-5)
DTLZ3		5.31e-3(6.27e-4)−	4.56e-3(6.74e-3)−	4.48e-2(4.91e-4)−	4.24e-3(1.71e-4)
DTLZ4		2.96e-2(2.54e-1)−	2.25e-1(3.44e-1)+	7.79e-2(2.25e-1)−	4.41e-3(7.58e-5)
DTLZ5		5.01e-3(1.38e-4)−	3.96e-3(1.31e-7)+	4.18e-3(3.94e-5)+	4.40e-3(5.66e-5)
DTLZ6		4.95e-3(1.76e-4)−	3.96e-3(3.71e-8)+	4.10e-3(3.61e-5)−	4.09e-3(1.83e-5)
DTLZ7		5.20e-3(1.56e-4)−	1.83e-1(2.17e-1)−	4.79e-3(8.62e-5)−	4.43e-3(4.01e-5)
WFG1	2	1.69e-1(4.68e-2)+	4.62e-1(5.99e-2)+	1.72e-1(5.29e-2)+	9.62e-1(6.74e-2)
WFG2		1.25e-2(6.34e-4)−	6.57e-2(3.59e-3)−	1.13e-2(4.42e-4)−	1.12e-2(2.33e-4)
WFG3		1.53e-2(7.71e-4)−	2.22e-2(2.53e-3)−	1.32e-2(6.35e-4)≈	1.32e-2(2.49e-4)
WFG4		1.56e-2(6.52e-4)+	2.75e-2(2.45e-3)+	1.37e-2(3.29e-4)+	3.88e-2(7.95e-3)
WFG5		6.56e-2(1.21e-3)−	7.16e-2(2.04e-3)−	6.52e-2(1.63e-3)−	6.46e-2(3.41e-4)
WFG6		8.02e-2(1.66e-2)−	1.09e-1(2.47e-2)−	7.61e-2(2.33e-2)−	1.72e-2(3.01e-3)
WFG7		1.69e-2(8.96e-4)−	2.63e-2(4.42e-3)−	1.38e-2(3.11e-4)≈	1.38e-2(2.38e-4)
WFG8		1.11e-1(1.40e-3)+	1.23e-1(7.68e-3)−	1.01e-1(2.15e-3)+	1.16e-1(1.78e-3)
WFG9		2.95e-2(3.79e-2)−	4.85e-2(2.99e-2)−	1.97e-2(2.25e-3)+	2.50e-2(2.07e-3)
DTLZ1	3	2.69e-2(1.06e-3)+	2.75e-3(8.28e-4)+	2.20e-2(1.14e-4)+	4.43e-2(7.83e-2)
DTLZ2		5.02e-3(2.88e-3)−	3.96e-2(1.02e-7)−	4.18e-3(5.02e-5)+	4.33e-3(5.84e-4)
DTLZ3		7.54e-2(7.43e-3)+	5.78e-2(7.41e-3)+	5.36e-2(7.12e-4)+	1.23e-0(9.68e-1)
DTLZ4		2.96e-2(1.34e-1)−	2.60e-1(2.71e-1)−	7.79e-2(2.25e-1)−	4.41e-3(7.58e-3)
DTLZ5		5.06e-3(1.76e-4)−	3.10e-2(9.54e-5)−	4.15e-3(3.47e-5)+	4.32e-3(3.97e-5)
DTLZ6		5.06e-3(2.09e-4)−	3.12e-2(3.10e-5)−	4.19e-3(3.01e-5)−	4.16e-3(2.49e-5)
DTLZ7		5.24e-3(1.66e-4)+	1.83e-1(2.17e-3)−	4.73e-2(3.22e-3)−	4.43e-2(4.43e-5)
WFG1	3	4.19e-1(5.37e-2)+	5.68e-1(7.88e-2)+	3.88e-1(6.14e-2)+	1.58e-0(2.50e-2)
WFG2		1.94e-1(8.65e-3)−	9.23e-1(1.38e-2)−	1.59e-1(9.27e-3)−	1.56e-1(7.49e-3)
WFG3		1.01e-1(1.37e-2)+	1.71e-1(5.01e-2)−	1.13e-1(9.28e-3)+	1.50e-1(6.12e-3)
WFG4		2.73e-1(1.05e-2)−	2.33e-1(3.67e-3)+	2.20e-1(2.43e-3)+	2.59e-1(3.61e-3)
WFG5		2.76e-1(8.95e-3)−	2.36e-1(2.06e-3)−	2.31e-1(3.49e-3)+	2.35e-1(2.49e-3)
WFG6		3.11e-1(1.54e-2)−	2.74e-1(1.49e-2)−	2.43e-1(1.21e-2)−	2.41e-1(7.72e-3)
WFG7		2.81e-1(1.12e-2)−	2.70e-1(2.18e-2)−	2.28e-1(3.32e-3)−	2.26e-1(3.20e-3)
WFG8		3.73e-1(9.08e-3)−	3.01e-1(7.08e-3)+	3.03e-1(4.77e-3)+	3.32e-1(5.33e-3)
WFG9		2.74e-1(1.12e-2)−	2.67e-1(3.52e-2)−	2.23e-1(2.33e-2)−	2.17e-1(2.24e-3)
+/−/≈		10/27/0	13/24/0	17/18/2	

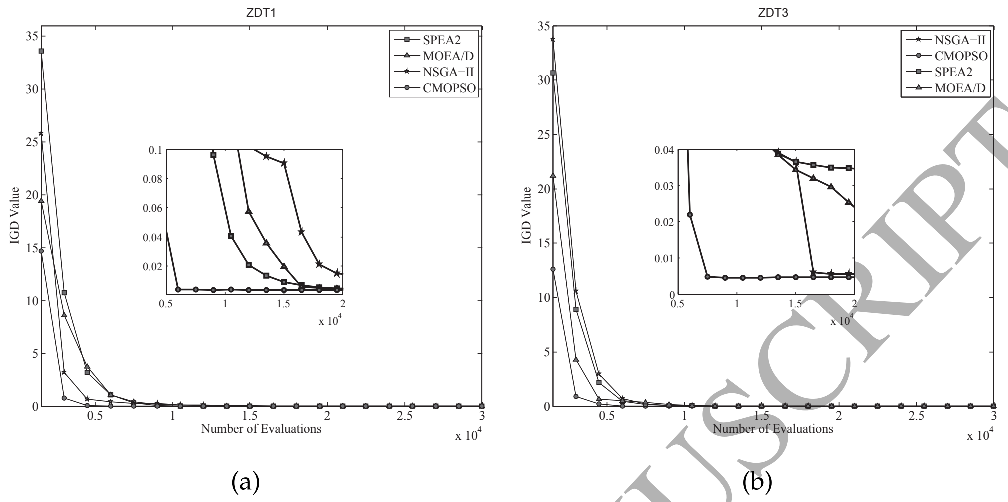


Figure 4: Convergence trajectories of the proposed CMOPSO and three MOEAs, NSGA-II, MOEA/D and SPEA2 on ZDT1 and ZDT3, averaging over 30 runs. (a) Convergence trajectory of the four compared multi-objective PSO algorithms on ZDT1; (b) Convergence trajectory of the four compared multi-objective PSO algorithms on ZDT3. The experimental results indicate the competitive convergence speed of the proposed CMOPSO in comparison with the three popular MOEAs on ZDT1 and ZDT3.

premature convergence due to the loss of swarm diversity, but a much large elite particle set, on the other hand, often leads to a lower learning efficiency and thus reduces the convergence speed. Therefore, we perform the sensitivity analysis of the parameter γ in this subsection.

Figure 7 presents the IGD values of the proposed CMOPSO with different sizes of elite particle set on DTLZ1, DTLZ2, DTLZ3 and DTLZ7 with two and three objectives, averaging over 30 runs, where DTLZ1 and DTLZ3 are two representative MOPs with local fronts, DTLZ2 has a simple continuous Pareto front, and DTLZ7 has a disconnected Pareto front. It can be observed that the performance of CMOPSO is relatively sensitive to the size of elite particle set on MOPs with local fronts, especially in the case of three objectives. For both bi-objective and three-objective DTLZ1 and DTLZ3, the proposed CMOPSO achieved the best performance when the size of elite particle set is set to 10. By contrast, the proposed CMOPSO demonstrates more robust performance to the settings of γ on DTLZ2 and DTLZ7, which are without local fronts. Therefore, a size of 10 is usually

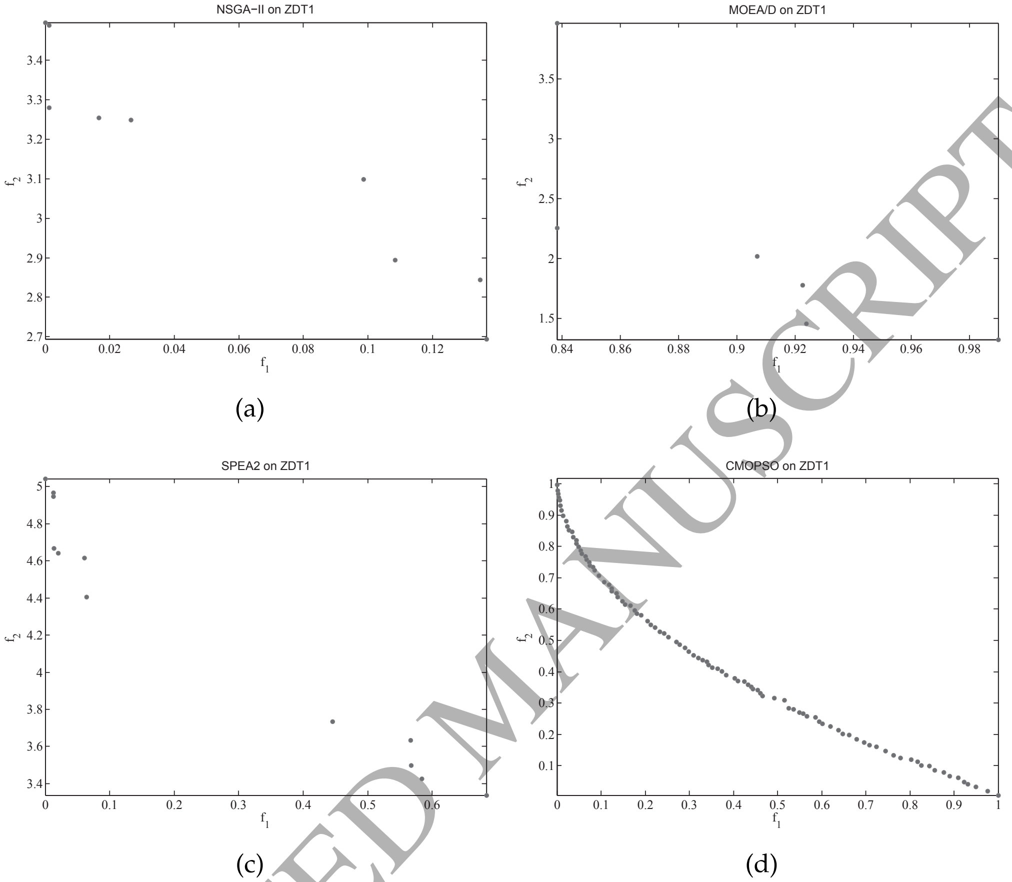


Figure 5: The non-dominated solution sets associated with the best IGD value among 30 runs obtained by the proposed CMOPSO and three MOEAs with a maximum number of generations of 50 on ZDT1. (a) The non-dominated solution set obtained by NSGA-II on ZDT1; (b) The non-dominated solution set obtained by MOEA/D on ZDT1; (c) The non-dominated solution set obtained by SPEA2 on ZDT1; (d) The non-dominated solution set obtained by CMOPSO on ZDT1. The experimental results show that the proposed CMOPSO can well converge to the whole Pareto front of ZDT1 within a small number of generations.

suggested for elite particle set in the proposed CMOPSO to solve MOPs.

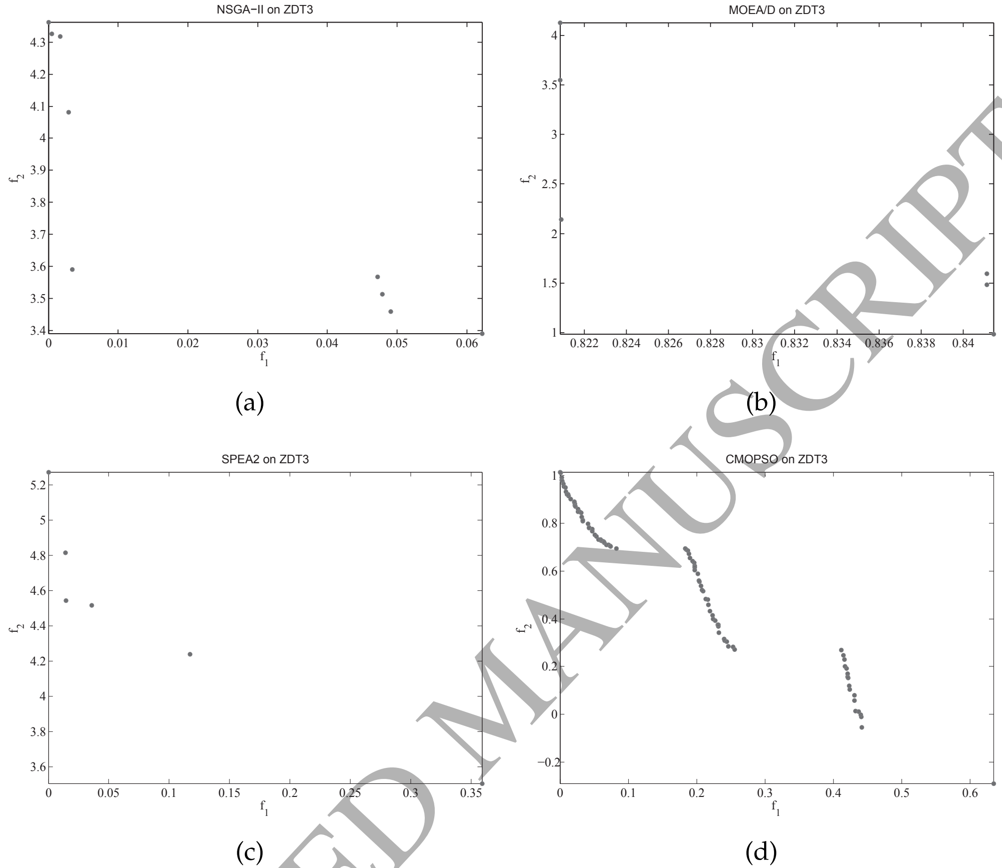


Figure 6: The non-dominated solution sets associated with the best IGD value among 30 runs obtained by the proposed CMOPSO and three MOEAs with a maximum number of generations of 50 on ZDT3. (a) The non-dominated solution set obtained by NSGA-II on ZDT3; (b) The non-dominated solution set obtained by MOEA/D on ZDT3; (c) The non-dominated solution set obtained by SPEA2 on ZDT3; (d) The non-dominated solution set obtained by CMOPSO on ZDT3. The experimental results show that the proposed CMOPSO can well converge to the whole Pareto front of ZDT3 in within a small number of generations.

5. Conclusion and remark

In this paper, we have proposed a competitive mechanism based multi-objective particle swarm optimizer (CMOPSO) inspired from the recently proposed competitive swarm optimizer [2]. In CMOPSO, a competition mechanism based learning strategy has been suggested for updating the

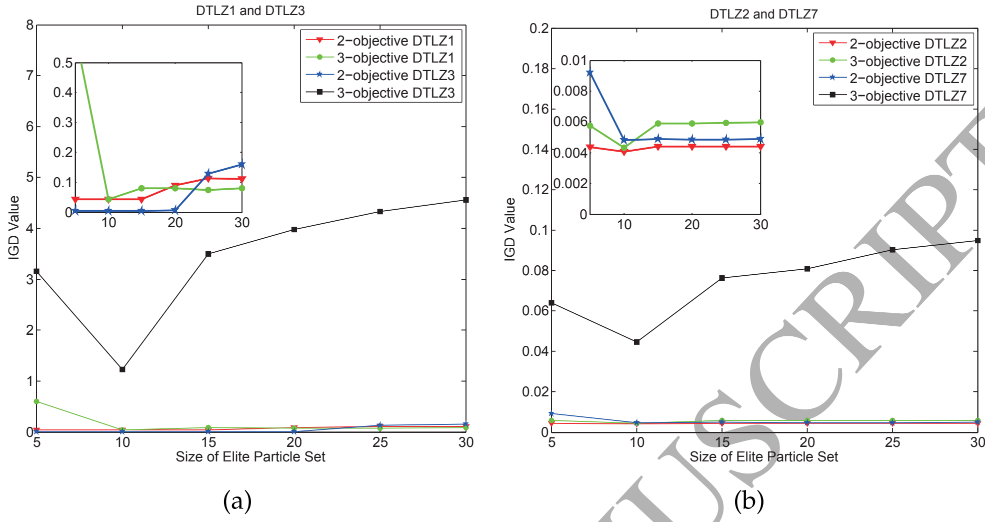


Figure 7: The IGD values of the proposed CMOPSO with different sizes of elite particle set on DTLZ1, DTLZ2, DTLZ3 and DTLZ7 with two and three objectives. (a) The IGD values of the proposed CMOPSO with different sizes of elite particle set on DTLZ1 and DTLZ3 with two and three objectives; (b) The IGD values of the proposed CMOPSO with different sizes of elite particle set on DTLZ2 and DTLZ7 with two and three objectives. The experimental results demonstrate that 10 is often a good setting for the size of elite particle set on DTLZ1 and DTLZ3.

particles, where each particle is made to learn from the winner of each pairwise competition. Since the competitions are performed among the elite particles selected from the current swarm, there is no external archive maintained for saving global or personal best particles. Experimental results on a variety of benchmark MOPs have demonstrated the promising performance of the proposed CMOPSO, in comparison with several existing multi-objective PSO algorithms and popular MOEAs.

In the future, it is interesting to investigate the pairwise competition mechanism by further exploring its potential in solving more complicated MOPs, e.g., those with a large number of local fronts [4]. In addition, the scalability of competition mechanism suggested in CMOPSO [3] to large-scale MOPs also deserves to be instigated, since competitive swarm optimizer has demonstrated its competitiveness in solving large-scale SOPs.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 61672033, 61502004, 61502001), and the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the National Natural Science Foundation of China (Grant No. 61428302).

References

[1] H.A. Abbass, R. Sarker, and C. Newton. A Pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 971–978, 2001.

[2] R. Cheng and Y. Jin. A competitive swarm optimizer for large scale optimization. *IEEE Transactions on Cybernetics*, 45(2):191–204, 2015.

[3] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. Test problems for large-scale multi-objective and many-objective optimization. *IEEE Transactions on Cybernetics*, 2016, in press.

[4] Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, 2017.

[5] C.A. Coello Coello and M.S. Lecsuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1051–1056, 2002.

[6] C. Dai, Y. Wang, and M. Ye. A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences*, 325(12):541–557, 2015.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multiobjective optimization test problems. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 825–830, 2002.

[9] I. Fister, P. N. Suganthan, I. Fister Jr, S.M. Kamal, F.M. Al-Marzouki, M. Perc, and D. Strnad. Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution. *Nonlinear Dynamics*, 84(2):895–914, 2016.

[10] J. Handl, D.B. Kell, and J. Knowles. Multiobjective optimization in bioinformatics and computational biology. *IEEE Transactions on Computational Biology and Bioinformatics*, 4(2):279–292, 2007.

[11] J.G. Herrero, A. Berlanga, and J. M. M. López. Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters. *IEEE Transactions on Evolutionary Computation*, 13(1):151–168, 2009.

[12] W. Hu and G.G. Yen. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on Evolutionary Computation*, 19(1):1–18, 2015.

[13] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[14] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics-Part C*, 28(3):392–403, 1998.

[15] Y. Jiang, T. Hu, C. Huang, and X. Wu. An improved particle swarm optimization algorithm. *Applied Mathematics and Computation*, 193(1):231–239, 2007.

[16] Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 1042–1049, 2001.

[17] I. Fister Jr, M. Perc, S.M. Kamal, and I. Fister. A review of chaos-based firefly algorithms: perspectives and research challenges. *Applied Mathematics and Computation*, 252:155–165, 2015.

[18] I. Fister Jr, M. Perc, K. Ljubi, S.M. Kamal, A. Iglesias, and I. Fister. Particle swarm optimization for automatic creation of complex graphic characters. *Chaos, Solitons and Fractals*, 73(4):29–35, 2015.

[19] K.Deb and S. Tiwari. Omni-optimizer: A procedure for single and multi-objective optimization. In *Proceedings of IEEE International Conference on Evolutionary Multi-Criterion Optimization*, pages 47–61, 2005.

[20] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[21] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[22] L. Li, W. Wang, and X. Xu. Multi-objective particle swarm optimization based on global margin ranking. *Information Sciences*, 375(1):30–47, 2017.

[23] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming. A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research*, 247(3):732–744, 2015.

[24] S. Zapotecas Martínez and C.A. Coello Coello. A multi-objective particle swarm optimizer based on decomposition. In *Proceedings of Annual Conference on Genetic and Evolutionary Computation*, pages 69–76, 2011.

[25] A.J. Nebro, J.J. Durillo, J. Garcia Nieto, C.A. Coello Coello, F. Luna, and E. Alba. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *Proceedings of IEEE Symposium on Computational Intelligence in Miulti-Criteria Decision-Making*, pages 66–73, 2009.

[26] A. Nickabadi, M.M. Ebadzadeh, and R. Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, 11(4):3658–3670, 2011.

[27] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM symposium on Applied Computing*, pages 603–607, 2002.

[28] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *Proceedings of IEEE International Conference on Granular Computing*, pages 534–537, 2008.

[29] A. Ponsich, A.L. Jaimes, and C.A. Coello Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, 2013.

[30] F. Samanlioglu. A multi-objective mathematical model for the industrial hazardous waste location-routing problem. *European Journal of Operational Research*, 226(2):332–340, 2013.

[31] R. Shang, L. Jiao, F. Liu, and W. Ma. A novel immune clonal algorithm for MO problems. *IEEE Transactions on Evolutionary Computation*, 16(1):35–50, 2012.

[32] M.R. Sierra and C.A. Coello Coello. Improving PSO-based multiobjective optimization using crowding, mutation and ϵ-dominance. In *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization*, pages 505–519, 2005.

[33] A. A. Taleizadeh, F. Barzinpour, and H. M. Wee. Metaheuristic algorithms for solving a fuzzy single-period problem. *Mathematical Computer Modelling*, 54(5-6):1273–1285, 2011.

[34] A. A. Taleizadeh, S. T. A. Niaki, and M. B. Aryanezhad. A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory control systems with random fuzzy replenishments. *Mathematical and Computer Modelling*, 49(5-6):1044–1057, 2009.

[35] A. A. Taleizadeh, S. T. A. Niaki, M. B. Aryanezhad, and A. F. Tafti. A genetic algorithm to optimize multiproduct multiconstraint inventory control systems with stochastic replenishment intervals and discount. *International Journal of Advanced Manufacturing Technology*, 51(1):311–323, 2010.

[36] A. A. Taleizadeh, S. T. A. Niaki, and N. Shafii. A hybrid method of fuzzy simulation and genetic algorithm to optimize constrained

inventory control systems with stochastic replenishments and fuzzy demand. *Information Sciences*, 220(3):425–441, 2013.

[37] A. A. Taleizadeh, S. T. A. Niaki, N. Shafii, R. G. Meibodi, and A. Jabbarzadeh. A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead time and (r, q) policy in supply chain. *International Journal of Advanced Manufacturing Technology*, 51(9-12):1209–1223, 2010.

[38] A. A. Taleizadeh, H. Shavandi, and R. Haji. Hybrid algorithms to solve constraint single period problem with uncertain demand. *Scientia Iranica*, 18(6):1553–1563, 2011.

[39] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin. An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, 2017, in press.

[40] Y. Tian, R. Cheng, X. Zhang, and Y. Jin. PlatEMO: A matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12(4):73–87, 2017.

[41] Y. Tian, H. Wang, X. Zhang, and Y. Jin. Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization. *Complex & Intelligent Systems*, 2017, in press.

[42] Y. Wang and Y. Yang. Particle swarm optimization with preference order ranking for multi-objective optimization. *Information Sciences*, 179(12):1944–1959, 2009.

[43] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. S. H. Chung, and Y. H. Shi. Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. *IEEE Transactions on Cybernetics*, 43(2):445–463, 2013.

[44] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1362–1381, 2009.

[45] J. Zhang, C. Zhang, T. Chu, and P. Matjaž. Resolution of the stochastic strategy spatial prisoner's dilemma by means of particle swarm optimization. *Plos One*, 6(7):e21787, 2011.

[46] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu. A mixed representation based multi-objective evolutionary algorithm for overlapping community detection. *IEEE Transactions on Cybernetics*, 47(9):2703–2716, 2017.

[47] Q. Zhang and H. Li. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[48] Q. Zhang, A. Zhou, and Y. Jin. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63, 2008.

[49] X. Zhang, F. Duan, L. Zhang, F. Cheng, Y. Jin, and K. Tang. Pattern recommendation in task-oriented applications: a multi-objective perspective. *IEEE Computational Intelligence Magazine*, 12(3):43–53, 2017.

[50] X. Zhang, Y. Tian, R. Cheng, and Y. Jin. An efficient approach to non-dominated sorting for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(2):201–213, 2015.

[51] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 892–899, 2006.

[52] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[53] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. In *Proceedings of IEEE Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2001.

[54] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: A comparative case study. In *Proceedings of International Conference on Parallel Problem Solving from Nature*, pages 292–301, 1998.