

Population recombination strategies for multi-objective particle swarm optimization

Li Ming Zheng¹ · Qiang Wang¹ · Sheng Xin Zhang¹ · Shao Yong Zheng²

Published online: 18 February 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Multi-objective particle swarm optimization algorithm (MOPSOs) has been found to exhibit fast convergence speed but with high probability to fall into local optimum. To overcome this shortcoming, a population recombination strategy is combined with a new mutation strategy to strengthen the ability to jump out of local optimum. From the investigation conducted, it can be found that, when the MOPSO falls into local optimum, the population will stop producing effective particles to update the archive. Population recombination strategy, which utilizes the information of the best variable found so far to construct the new population. This can increase the probability for population to approach the Pareto optimal front, while additional mutation operation can enhance the diversity of population. Experimental study on the bi-objective and three-objective benchmark problems shows that the MOPSO based on proposed strategies is superior to previous multi-objective algorithms in the literature.

Keywords Multi-objective particle swarm optimization · Population recombination strategies · Mutation strategy · Local optimum

Communicated by V. Loia.

✉ Shao Yong Zheng
xmuzsy@hotmail.com

Li Ming Zheng
zlmqit@126.com

¹ Department of Electronic Engineering, School of Information Science and Technology, University of Jinan, Guangzhou 510632, China

² Department of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510006, China

1 Introduction

Since the real-world problems need to optimize multiple conflicting objectives simultaneously (Tate et al. 2012), multi-objective optimization problems (MOPs) get widespread attention in recent decades. The MOPs need to find the optimal solution set named Pareto optimal Front (*PF*) that consists of multiple solutions. As Evolutionary Algorithms (EAs) are population-based algorithms which can find multiple solutions in a run, researchers try to extend it to solve the MOPs. Thus, multi-objective evolutionary algorithms (MOEAs) get intensive and extensive study (Deb 2001; Xia et al. 2014).

Particle swarm optimization algorithm (PSO) (Kennedy and Eberhart 1995; Shi and Eberhart 1998) is a population-based algorithm that simulates the movements of a flock of birds to find food. Because of the fast convergence and simple implementation, PSO is also extended to solve MOPs (Margarita et al. 2006) (named MOPSO), and successfully applied in dealing with many real-world problems (Lalwani et al. 2013; Xue et al. 2013). Compared with other algorithms (such as MOEAs), MOPSO has unique particle update method and personal best (*pbest*) to record particle experiences. In fact, there are two types of archives (Coello et al. 2004) in MOPSOs, one is composed of global optimum (*gbest*) and another is composed of all of the *pbest* which are recorded by particles.

However, similar to other multi-objective optimization algorithms, MOPSOs also encounter the problem of premature convergence for the complex MOPs. Many researchers noticed this problem, and learned from Genetic Algorithm (GA) which utilizes a mutation operation to enhance the ability to avoid the local optimum, thus the mutation operation was introduced in the PSOs and MOPSOs as well (Paul 2006). Other researchers try to combine variety of algorithms

to solve MOPs (Luis et al. 2006; Daneshyari and Yen 2011). Besides, the comprehensive learning PSO (CLPSO) (Liang et al. 2006) aims to enhance the diversity of the population to improve the algorithm's ability to escape from local optima. Based on this, the Multi-objective comprehensive learning particle swarm optimizer (MOCLPSO) (Huang et al. 2006) was proposed to solve MOPs with excellent performance.

For the simple and effective characteristics, most of MOPSOs use the Pareto relationship to update $gbest$ and $pbest$. Moore and Chapman (1999) utilized a ring topology to solve the MOPs problems. In the proposed topology, the $pbest$ of a particle is a list of all non-dominated solutions found in its trajectory. In the process of updating the velocity and position, one $pbest$ is selected randomly from the list. Li (2003) proposed the non-dominated sorting particle swarm optimizer (NSPSO) that incorporates the main mechanisms of the NSGA-II (Deb et al. 2002). In NSPSO, once a particle has updated its position, all the $pbest$ of the particles and all the new positions obtained recently are combined as a solution set and sorted into various non-domination levels using the non-dominated sorting technology. Niche count and crowding distance are used to estimate the density. N (population size) best particles are selected as the new swarm. This approach applies a mutation operator only on the particle with the smallest crowding distance value (or the largest niche count). Mostaghim and Teich (2003) proposed sigma-MOPSO, in which the $gbest$ is selected according to the closest sigma value. Speed-constrained Multi-objective PSO (SMPSO) (Nebro et al. 2009) is an improved version of OMOPSO (Sierra and Coello 2005), which uses an adaptive grid technology to update external archive and utilizes roulette-wheel method to select $gbest$. In MOCLPSO, the $gbest$ is selected randomly from archive, but the $pbest$ is selected from different particles of swarm to enhance the diversity of algorithm. The co-evolutionary multi-swarm PSO (CMPSO) (Zhan and Zhang 2013) introduced a multiple populations for multiple objectives (MPMO) technology with each objective corresponding to a population. The elitist learning strategy (ELS) is also applied in the archive to strengthen the ability of exploring.

Inspired by the MOEA/D (Zhang and Li 2007) which decomposes the original MOP into a number of scalar aggregation problems, Moubayed et al. (2010) proposed a novel smart MOPSO using decomposition (SDMOPSO), which associates every particle with a vector according to the best scalar aggregated fitness value and utilizes archival storage of non-dominated solutions. Hybrid technologies had been also applied to enhance the MOPSOs. In literature (Tang and Wang 2013), the hybrid multi-objective evolutionary algorithm (HMOEA) was reported. A self-adaptive selection mechanism is developed to choose an appropriate crossover operator from several candidates.

The MOPSOs exhibits a strong ability to learn but with weakness in exploration. Assume that there is only one global optimum, particles in the population will soon fly to the global optimum and begin to explore new areas. As the population concentrates to a small area, the ability to explore new areas becomes weaker. MOPSOs can benefit from the introduction of archive which contains multiple global optimum, but exploring ability has not been strengthened.

Through the investigation on the size of archive, it can be found that it will soon increase to the maximum value for simple problems. However, the archive size for complex functions tends to be small and remains unchanged over a period of time. For verification, Fig. 1a shows the trend for the archive size (na) for the ZDT1 and ZDT4 issues obtained by the conventional PSO algorithm. As can be seen from the

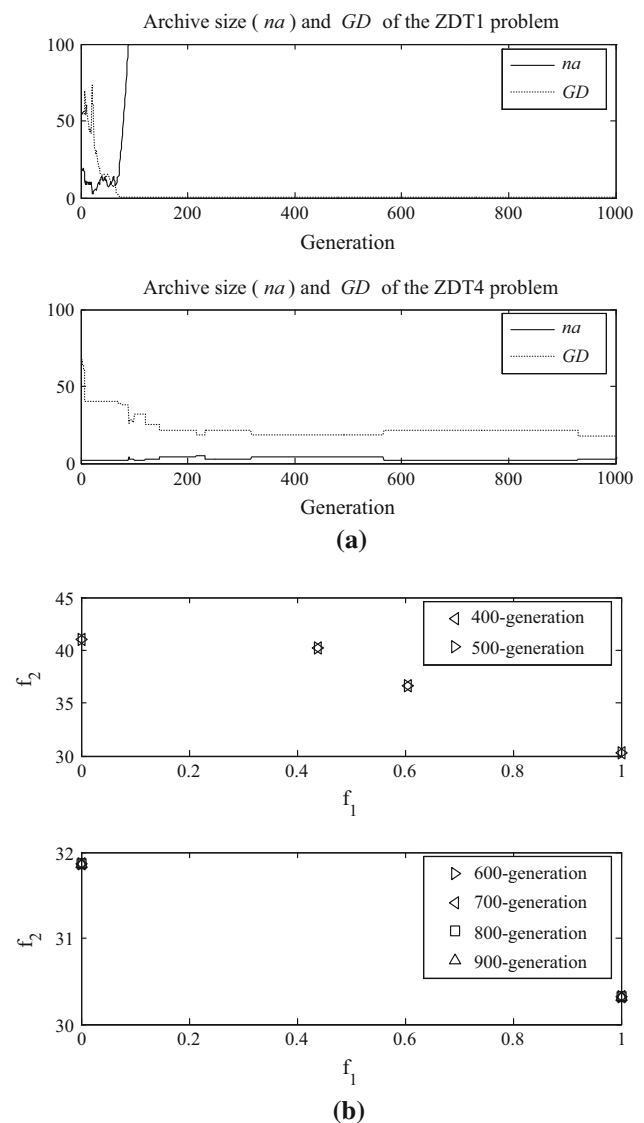


Fig. 1 a Archive size and convergence curve of the ZDT1 and ZDT4 problems. For ZDT1 problem, to facilitate the observation, the convergence value multiplied by 100. b The distribution of each generation of the archive for ZDT1 problem

figure, the archive size for ZDT1 problems quickly increases to the limit of archive ($NA = 100$) and remains unchanged. For ZDT4 problem, the archive size does not reach NA , and maintains a very small value. Furthermore, there is no change in the size of the archive (such as 400 and 500 generations), and the convergence value (generational distance indicator GD) of the algorithm did not change during this period. To observe the results intuitively, the solutions in the archive of 400, 500, 600, 700, 800, 900 generations are also shown. Figure 1b shows the distribution of archive for 400–500 and 600–900 generation. It can be observed the archive remains unchanged, because there is no new particle added to the archive when the archive is updated. There exist two cases: one is the new particle does not dominate any solution in the archive, this would be possible to make the archive size smaller. The other is the population did not generate any particle that non-dominates any solution in archive, which will increase the size of the archive. This trend should be a manifestation of complex problems, the population cannot produce effective particle to update the archive.

To overcome this problem, the population recombination strategy (PRS) and a new mutation strategy are proposed. The main idea is to discard most of current position information which is useless for generating better solution, and effectively use the information contained in the archive and all of the $pbest$ to construct a new particle. Moreover, the mutation operator is introduced to enhance the algorithm's ability to escape from local optimum. PRS was implemented using the best variables information found to construct the new population, which can increase the probability of particles to be included in the archive. At the same time, the mutation operation is also combined to strengthen the ability to jump out of local optimum. The detailed description about PRS and new mutation strategy will be described in the paper. Based on the PRS and new mutation strategies, a multi-objective particle swarm optimization algorithm called RMOPSO is proposed and verified.

The novelties of proposed RMOPSO algorithm lie in:

1. Different from existing algorithms in which the swarm is updated continuously, the PRS recombines the swarm in a certain stage to discard some useless information which is likely to mislead the direction of evolution. The chance for the swarm to regroup, therefore, increases.
2. Different from existing algorithms that mutate particle by probability or the value of density like the NSPSO, the new mutation strategy aims to mutate particles that had been included in the archive. Therefore, it becomes more effective when the algorithms fall into local optimum.

The rest of this paper is organized as follows. Section 2 constructs a MOPSO framework based on the predecessors' research and describes the PRS and new mutation strategy

in detail. The experimental analysis of proposed strategy is presented in Sect. 3. Section 4 reports and analyzes the experimental results on benchmark problems. The comparison with other state-of-the-art MOEAs and MOPSOs is also conducted. Finally, some conclusions are drawn in Sect. 5.

2 RMOPSO

2.1 Basic framework of RMOPSO

Based on previous studies, RMOPSO which utilizes the Pareto relationship to evaluate the quality of the solutions is constructed. The basic framework is composed of 3 processes, which are swarm initialization process, swarm update process and archive update process. The detailed pseudo-code can be found in Fig. 2.

In Swarm_update_process described in Fig. 2b, the updated velocity and position may exceed their corresponding boundary, they should be adjusted to meet the velocity constraints V_i and position constraints X_i after the update process:

- Case 1: if $V_i > V_{max,i}$, then set V_i as $V_{max,i}$.
Case 2: if $V_i < V_{min,i}$, then set V_i as $V_{min,i}$.

Here $V_{max,i} = \theta X_{max,i}$ and $V_{min,i} = \theta X_{min,i}$. Literature (Zhan et al. 2009) suggests that $\theta = 0.2$ ($V_{max,i} = 0.2X_{max,i}$ and $V_{min,i} = 0.2X_{min,i}$). However, it was found that $\theta = 1.0$ can improve the diversity of the algorithm, thus 1.0 is used in this paper. For the position constraints are given as follow:

- Case 1: if $X_i > X_{max,i}$, then $X_i = X_{max,i}$ and $V_i = -V_i$.
Case 2: if $X_i < X_{min,i}$, then $X_i = X_{min,i}$ and $V_i = -V_i$.

The archive denoted as A is adopt to store all non-dominated solutions found so far; it is initialized to be empty and updated in every generation. Research shows that it is better to use an archive with a fixed maximum size, as the number of non-dominated solutions may increase very fast. Therefore, the maximum size of the archive is denoted as NA and the current size is denoted as na . In the update process of archive, all the particles in swarm and solutions in archive are combined as a set firstly, and all particles dominated by any other particles are deleted. The density of each solution is calculated and sorted using the crowding distance technology. If the index value is larger than the maximum limit NA , the corresponding solution will be deleted.

2.2 Population recombination strategy

The pseudo-code of the PRS is given in Fig. 3. The position information of current particle is not completely useless and

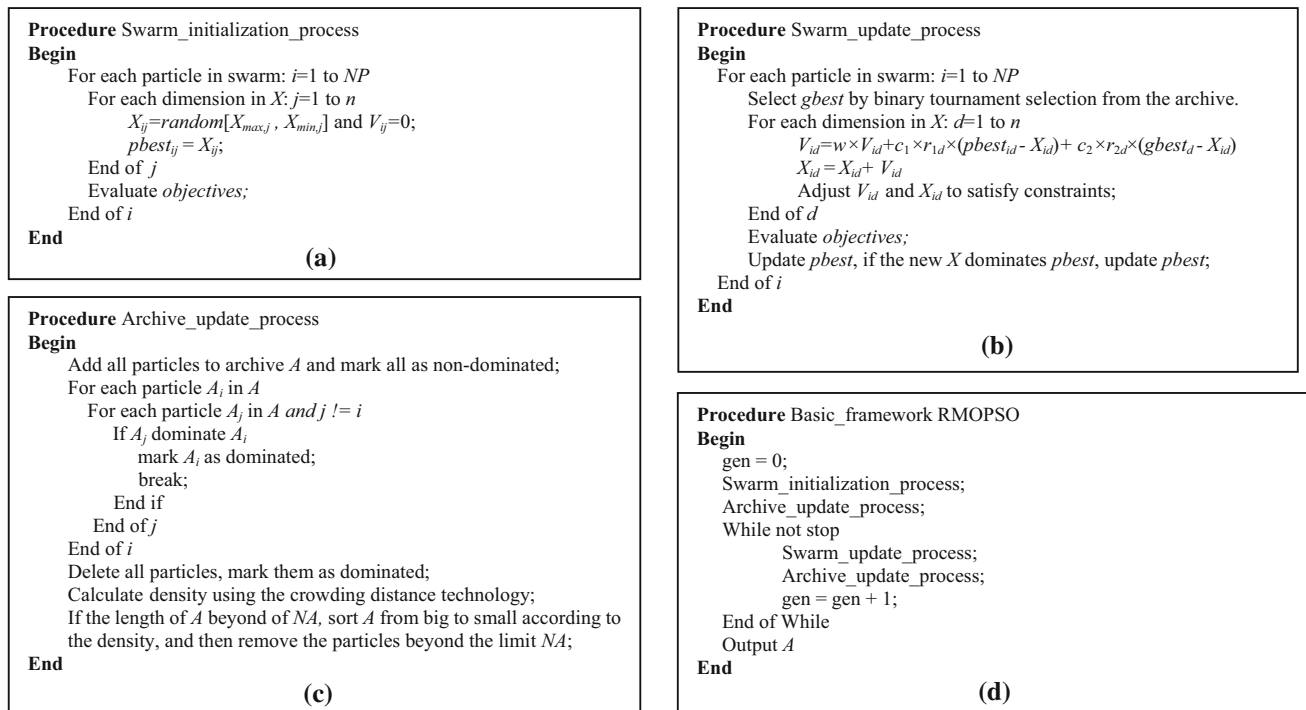


Fig. 2 The pseudo-codes of the basic RMOPSO, **a** pseudo-code of the swarm initialization process; **b** pseudo-code of the swarm update process; **c** pseudo-code of the archive update process; **d** pseudo-code of the basic framework of the RMOPSO

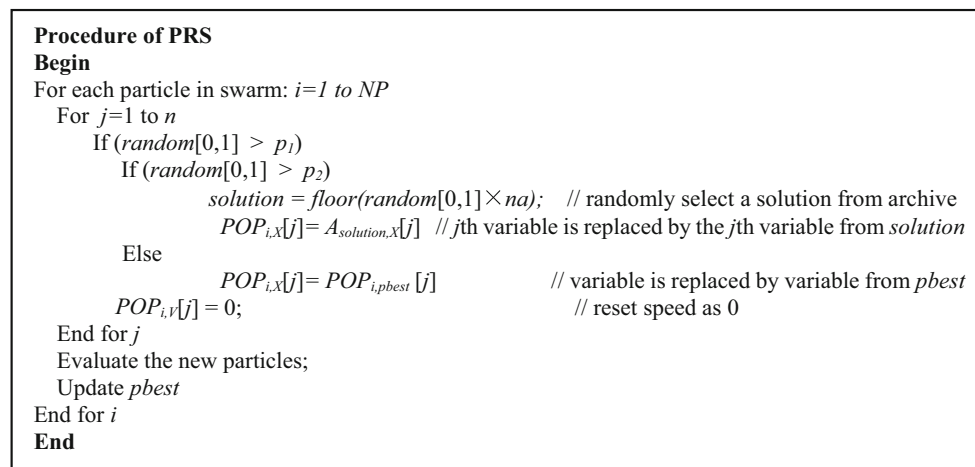


Fig. 3 The pseudo-code of the PRS. $random[0,1]$ is used to generate a random number between 0 and 1, and A denotes the archive

PRS do not discard all of the position information belong to the particles, it will be replaced by the $pbest$ or $gbest$ based on the probability p_1 for each dimension.

As a storage pool, the archive saves all optimal solutions found so far. However, when the algorithms fall into a local optimum, the stored information is not reliable. $pbest$ is the best position that each particle in swarm discovers and records. In most cases, $pbest$ should also be included in the archive, but when the archive size is smaller than the size of population, obviously, some $pbest$ of the particle in swarm is significantly different from the solutions in archive. Thus, all

of the $pbest$ have some diversity information of the swarm. Therefore, all of the information stored in archive and $pbest$ in swarm are important and should be fully utilized. However, it is difficult to know whether the information is valid in the process. Thus, the probability p_2 is introduced to determine the use of $pbest$ and $gbest$.

After PRS, the velocity is set as 0. It should be noted that the PRS utilizes variables information directly and does not create any new position information and the $pbest$ which belongs to each recombined particle is unchanged.

A problem arises when the *PRS* is used to construct new swarm. The *PRS* cannot be used in every generation, because it cannot create any new position information but involves adjustment based on existing variables. Therefore, a parameter called generation interval T is introduced, which means the *PRS* is used for every T generation. In the Sect. 3, experimental analysis on this parameter is conducted and the results show that the algorithm exhibits better performance when T is chosen between 5 and 10.

2.3 New mutation strategy

As the mutation operator can enhance the diversity of population and strengthen the exploration ability of the algorithm. It was introduced in the PSO and MOPSO by researchers. Similar to the MOEAs, the probability for the mutation of one particle is $1/n$, where n is the number of variables. Besides, MOPSO uses a mutation operation whose probability decreases with increasing iterations. While the NSPSO mutates the particles which has a smallest density value.

The mutation operator is also employed here. During the archive update process, some particles will be added to the archive if these particles dominate or non-dominate solutions in archive. When updating the population, especially when the algorithms fall into local optimum, some particles in swarm are the global optimum and located in archive, the $pbest$ will choose $agbest$ from the archive to learn. However, they have the same level, and the effective of explore variable space is reduced. Accordingly, mutation of these particles is a good choice. When the particle is selected for mutation, a random dimension d is selected to perform Gaussian perturbation as:

If $random[0, 1] > 0.5$

$$POP_i[d] = POP_i[d] + (X_{max,d} - X_{min,d}) * gasussian(0, 1)$$

else

$$POP_i[d] = POP_i[d] - (X_{max,d} - X_{min,d}) * gasussian(0, 1)$$

where i is particle index selected to mutation, $X_{max,d}$ and $X_{min,d}$ are the upper and lower bounds of the d th dimension, respectively. $gasussian(0,1)$ is a random value generated by a Gaussian distribution with a mean value of 0 and a standard deviation of 1.

2.4 Population recombination multi-objective particle swarm optimization algorithm

Based on the *PRS* and new mutation strategy, Population Recombination Multi-objective particle swarm optimization

Algorithm RMOPSO

Begin

Input: 1) Size of population NP , maximum size of archive NA and current size $na = 0$.

2) Parameters: w, c_1, c_2 .

3) Position constraints and Velocity constraints:

$$X_i \in [X_{max,i}, X_{min,i}], V_i \in [V_{max,i}, V_{min,i}].$$

4) Stopping criterion: maximum number of function evaluation FES and set $count = 0$ which is used to record number of the function evaluation.

Output: archive of the last generation.

Initialization:

Initialize the swarm by Swarm_initialization_process;

Update archive by Archive_update_process;

End Initialization

Iteration:

if $t \% T = 0$

use *PRS* to recombination swarm;

else Update particles:

For $i=1$ to NP

Select $gbest$ from archive;

Update the velocity and position;

//Mutation operator;

If $POP_{i,min-distance} = 0$, mutation this particles.

Evaluate the new particles and $count+1$.

Update $pbest$.

End of i

2).Update archive:

Find all non-dominated solutions from the union set of swarm and archive, then truncate it when $na > NA$. In this process, if i th particle in the current swarm is added into the archive, mark it using $POP_{i,min-distance} = 0$.

End Iteration:

Criterion:

If $count > FES$, then output archive and stop, otherwise go to Iteration.

END

Fig. 4 The pseudo-code of complete RMOPSO algorithm

algorithm (RMOPSO) is implemented. The detailed flowchart is described in Fig. 4. In the flowchart, t is the number of generation of the RMOPSO, T is the Generation Interval, and $t \% T$ is the modulo operation used to determine whether the *PRS* process will be utilized. The swarm update process is same as the Swarm_update_process described in Fig. 2b except the new mutation operation.

The archive update process is similar to the Archive_update_process, as described in Fig. 2c. The particles included in the archive from swarm are marked for further mutation operation. The $count$ is a variable used to record the number of the function evaluations (FES).

3 Experimental analysis

To verify the effectiveness of new mutation strategy and *PRS* described in Sects. 2.2 and 2.3, an experimental analysis was carried out in this section. It includes analysis on different generation interval T for *PRS*, and the impacts of *PRS* and mutation operator. The basic framework of the RMOPSO described in Fig. 2d without the *PRS* and mutation operation will be used as the baseline algorithm in this section to demonstrate the advantages of the *PRS* and mutation operation.

3.1 Test problems

Six benchmark problems are adopted as test problems in this section. Among them, the problems (ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6) are selected from the ZDT test set, and the WFG1 is selected from the WFG test set (Huband et al. 2005). ZDT1 and ZDT2 are simple problems and their *PF* is convex and concave, respectively, while the *PF* of ZDT3 is convex and disconnected. WFG1 has a mixed *PF*. ZDT4 is multimodal problems that have local optimum, thus it is used to test the ability of algorithm to jump out of the trap. These test problems consider various characteristics of the MOP and can be used to evaluate the performance of proposed algorithm regarding different issues.

3.2 Performance metric

In this section, the generational distance (*GD*) and Δ indicators are used to evaluate the convergence and diversity performance, respectively. The detailed definition can be found in Appendix 1. As these indicators require sufficient reference points to simulate the *PF*, 500 reference points are generated based on the known *PF* for each problem.

3.3 The impact of generation interval *T*

In this subsection, the impact of *T* on the RMOPSO is studied. The related parameter setting of the RMOPSO is: $NP = 50$, $NA = 100$, $w = 0.729$, $c_1 = c_2 = 2.05$,

$p_1 = 0.05$, $p_2 = 0.5$. In the experiment, each algorithm is run independently for 30 times and the maximum number of function evaluations (*FEs*) is 25,000 (If there are no special instructions, in the subsection 3.4, this setting will be also applied). The *T* is set to 100, 50, 20, 10, 7, 5 and 3, respectively, for comparison, $T = 0$ means that the *PRS* has not been used, which refers to the original MOPSO.

Table 1 shows the results for the RMOPSO with different values of *T*. The result shows that for convergence, with the decrease of *T*, convergence value significantly improves, especially for ZDT4 which is a complex problem. Overall, with the decrease of *T*, standard deviation of the value decreases as well, which indicates that the probability for the algorithm to converge to the optimal front increases. It is worthy to mention that for simple problems, the corresponding convergence values for $T = 10, 7, 5$ and 3 have the trend to decrease, but the differences between them are not significant. While for the complex problems (ZDT4 problem), the corresponding differences are still large. Thus, for the complex problems, the proposed algorithm is superior favorable to jump out of local optimum with smaller *T*.

The results of the diversity performance are summarized in Table 1. It can be observed that diversity value also decreases with decreasing *T*, but it is different from the convergence values (ZDT4 excluded) which decrease monotone with the decreasing *T*. When $T = 10$, diversity performance is found to be the best. However, for ZDT4 and DTLZ1 diversity value also reduces with decreasing *T*, and achieves the best when $T = 3$.

Table 1 Generational distance (*GD*) and the diversity performance (Δ) of RMOPSO with different values of *T*

	<i>T</i>	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	WFG1
<i>GD</i>							
	$T = 0$	(MeanStd) 0.2376 _{1.47E-3}	0.5487 _{2.46E-02}	0.1534 _{8.04E-04}	26.4101 _{2.41E+01}	0.7635 _{5.81E-02}	0.0321 _{8.97E-06}
	$T = 100$	(MeanStd) 0.0984 _{9.20E-04}	0.0877 _{3.07E-03}	0.0823 _{8.01E-04}	8.0169 _{5.45E+00}	0.0434 _{9.00E-04}	0.0168 _{8.73E-06}
	$T = 50$	(MeanStd) 0.0387 _{1.28E-04}	0.0142 _{2.07E-04}	0.0288 _{2.24E-04}	3.4936 _{3.70E+00}	0.0014 _{2.36E-07}	0.0085 _{2.77E-06}
	$T = 20$	(MeanStd) 0.0036 _{1.60E-06}	0.0012 _{4.40E-07}	0.0039 _{3.74E-06}	0.4114 _{1.05E-01}	0.0012 _{1.16E-07}	0.0025 _{1.23E-07}
	$T = 10$	(MeanStd) 0.0011 _{5.70E-08}	0.0004 _{9.73E-09}	0.0017 _{7.79E-08}	0.0464 _{1.18E-03}	0.0011 _{8.31E-08}	0.0017 _{5.76E-08}
	$T = 7$	(MeanStd) 0.0008 _{6.77E-08}	0.0003 _{8.55E-10}	0.0016 _{4.12E-08}	0.0212 _{3.07E-04}	0.0012 _{6.67E-08}	0.0017 _{4.51E-08}
	$T = 5$	(MeanStd) 0.0007 _{8.00E-08}	0.0003 _{1.46E-09}	0.0016 _{7.42E-08}	0.0047 _{2.95E-04}	0.0014 _{4.98E-07}	0.0016 _{5.31E-08}
	$T = 3$	(MeanStd) 0.0008 _{1.09E-07}	0.0003 _{1.04E-09}	0.0017 _{7.41E-08}	0.0009 _{1.33E-05}	0.0013 _{2.89E-07}	0.0014 _{6.08E-08}
Δ							
	$T = 0$	(MeanStd) 0.9065 _{6.43E-03}	0.8326 _{1.15E-03}	0.9365 _{6.70E-03}	0.9628 _{1.93E-03}	1.1363 _{1.44E-02}	0.6038 _{1.12E-02}
	$T = 100$	(MeanStd) 0.8818 _{1.96E-02}	0.9109 _{1.48E-02}	0.9144 _{1.14E-02}	0.9596 _{9.80E-03}	1.1884 _{5.44E-02}	0.6272 _{4.43E-03}
	$T = 50$	(MeanStd) 0.8629 _{1.63E-02}	0.7465 _{5.76E-02}	0.9198 _{5.69E-03}	0.8759 _{7.90E-04}	0.3584 _{1.44E-03}	0.5998 _{4.24E-03}
	$T = 20$	(MeanStd) 0.7707 _{8.71E-03}	0.5261 _{4.67E-02}	0.9961 _{2.70E-03}	0.8404 _{5.88E-03}	0.3213 _{9.37E-04}	0.4551 _{2.64E-03}
	$T = 10$	(MeanStd) 0.2958 _{4.57E-04}	0.2971 _{1.20E-03}	0.7862 _{8.80E-04}	0.6624 _{2.97E-02}	0.3572 _{4.00E-04}	0.3020 _{4.13E-04}
	$T = 7$	(MeanStd) 0.3095 _{8.73E-04}	0.3102 _{5.27E-04}	0.7327 _{1.13E-03}	0.5134 _{1.07E-02}	0.3812 _{1.07E-03}	0.3059 _{2.52E-04}
	$T = 5$	(MeanStd) 0.3252 _{7.15E-04}	0.3474 _{1.02E-03}	0.7128 _{9.53E-04}	0.3344 _{1.67E-02}	0.3789 _{7.68E-04}	0.3191 _{6.04E-04}
	$T = 3$	(MeanStd) 0.3641 _{5.09E-04}	0.4042 _{2.42E-03}	0.7420 _{6.04E-05}	0.3228 _{1.32E-03}	0.3986 _{1.20E-03}	0.3386 _{7.19E-04}

From the convergence and diversity results, it can be concluded that for the convergence of the MOPs, the smaller T can result in better performance for simple problems, and the smaller T (less than 5) affects diversity performance insignificantly. Hence, to improve the convergence performance without causing deterioration in diversity, the interval T is recommended to be between 5 and 10.

3.4 The impacts of the PRS and new mutation strategy

The effects of T had been studied in previous section; it is also necessary to investigate the impact of the PRS and new mutation strategy separately compared to the original MOPSO. As the supplement to the proposed PRS, the purpose of the new mutation strategy is to strengthen the diversity of population, and thereby enhance the ability of the algorithm to avoid local optimum. To analyze the impact of mutation operation independently, the related operation in the RMOPSO is deleted.

Therefore, the results obtained with mutation operator eliminated can be employed to investigate the effects of PRS, and the results given in Sect. 3.3 can be used to show the effects of new mutation strategy.

Table 2 shows the convergence and diversity values with only PRS employed. The results in Table 2 show that: (1) for convergence performance, with decreasing T for the simple problems, the convergence value becomes smaller. While for the complex problems ZDT4, although they exhibit the same trend for smaller value, but the results obtained do not fully converge to the PF until $T = 3$. (2) For diversity performance, the trend is similar. However, the best values obtained when $T = 3$ (for the ZDT6, the best value is obtained when

$T = 7$) are always larger than the corresponding results in the Table 1. This is because the PRS does not involve variable information for population and cannot enhance diversity performance.

Therefore, it can be concluded that the PRS can strengthen the convergence performance of the MOPSO, but for the diversity performance, the effect is not significant.

Figure 5 shows the comparison between the results in Tables 1 and 2. In the figure, “PRS + Mutation” denotes the results obtained by the algorithm with the combination of PRS and the new mutation strategy, while the “only PRS” denotes the results obtained by algorithm with PRS only. The comparison shows that: (1) in terms of convergence, for ZDT1, ZDT2, ZDT3 and ZDT4 problems, convergence performance is better for the case using PRS only when $T > 20$, and the case with the combination of PRS and mutation has better performance when $T < 20$. For the ZDT6 and WFG1 problems, the case with the combination of PRS and mutation has better performance. However, it should be noticed that the best value is obtained when $T < 20$, which is similar to the case for the simple problems. (2) With the mutation operation employed, the diversity performance is better than that of approach with only PRS used. As can be seen from the above comparison, the mutation operation can improve the diversity of population and strengthen the algorithm’s ability to jump out of local optimum simultaneously. It can be concluded from the above results that:

1. PRS can effectively improve the convergence performance of the algorithm, because it utilizes the information of individual optimal and global optimal variable to

Table 2 The generational distance (GD) and diversity performance (Δ) of RMOPSO for different values of T with population recombination only

	T	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	WFG1
<i>GD</i>							
	$T = 100$	(MeanStd) 0.0571 _{8.02E-05}	0.0303 _{2.73E-04}	0.0331 _{1.94E-04}	2.4851 _{3.82E+00}	0.0708 _{2.62E-03}	0.0191 _{7.29E-06}
	$T = 50$	(MeanStd) 0.0367 _{6.94E-05}	0.0118 _{9.76E-05}	0.0183 _{2.79E-05}	1.0474 _{1.69E-01}	0.0281 _{9.76E-04}	0.0135 _{4.23E-06}
	$T = 20$	(MeanStd) 0.0099 _{4.23E-06}	0.0041 _{6.73E-06}	0.0064 _{5.69E-06}	1.5377 _{3.32E+00}	0.0019 _{3.48E-06}	0.0075 _{9.93E-07}
	$T = 10$	(MeanStd) 0.0055 _{1.34E-06}	0.0028 _{2.99E-06}	0.0029 _{4.84E-07}	0.4729 _{3.61E-02}	0.0012 _{1.81E-07}	0.0060 _{8.01E-07}
	$T = 7$	(MeanStd) 0.0037 _{2.01E-06}	0.0008 _{7.15E-07}	0.0026 _{4.02E-07}	0.3351 _{2.19E-02}	0.0014 _{5.21E-07}	0.0045 _{1.90E-06}
	$T = 5$	(MeanStd) 0.0034 _{6.80E-07}	0.0004 _{1.17E-07}	0.0022 _{2.07E-07}	0.4491 _{5.81E-02}	0.0013 _{1.46E-07}	0.0029 _{1.06E-06}
	$T = 3$	(MeanStd) 0.0014 _{1.10E-07}	0.0006 _{4.76E-08}	0.0017 _{1.09E-07}	0.5826 _{1.25E-01}	0.0012 _{1.21E-07}	0.0018 _{8.27E-08}
<i>Δ</i>							
	$T = 100$	(MeanStd) 1.4306 _{1.55E-03}	1.3991 _{4.55E-02}	1.4612 _{7.40E-04}	1.0111 _{3.05E-04}	1.4423 _{2.64E-03}	1.4251 _{3.58E-03}
	$T = 50$	(MeanStd) 1.5193 _{3.36E-03}	1.4220 _{7.72E-02}	1.4342 _{7.41E-04}	1.0122 _{1.34E-03}	1.4087 _{2.07E-02}	1.3647 _{1.87E-03}
	$T = 20$	(MeanStd) 1.4096 _{5.30E-03}	1.2499 _{3.00E-02}	1.3337 _{1.10E-03}	1.1102 _{6.35E+00}	0.7353 _{3.54E-02}	1.2315 _{2.16E-03}
	$T = 10$	(MeanStd) 1.1627 _{3.27E-03}	1.1192 _{1.33E-02}	1.1643 _{6.84E-04}	1.0027 _{6.44E-05}	0.4359 _{1.15E-03}	1.0921 _{3.80E-03}
	$T = 7$	(MeanStd) 0.9597 _{1.46E-02}	0.9881 _{8.33E-04}	1.1080 _{4.37E-04}	0.9811 _{5.47E-04}	0.4703 _{2.37E-03}	0.9840 _{1.85E-02}
	$T = 5$	(MeanStd) 0.9035 _{3.07E-02}	0.9803 _{3.48E-03}	1.0396 _{4.47E-03}	0.9411 _{1.91E-03}	0.4788 _{1.12E-03}	0.7926 _{9.72E-03}
	$T = 3$	(MeanStd) 0.6350 _{5.65E-03}	0.8263 _{3.20E-02}	0.9002 _{7.31E-04}	0.9683 _{2.36E-03}	0.5350 _{1.42E-03}	0.7221 _{6.02E-03}

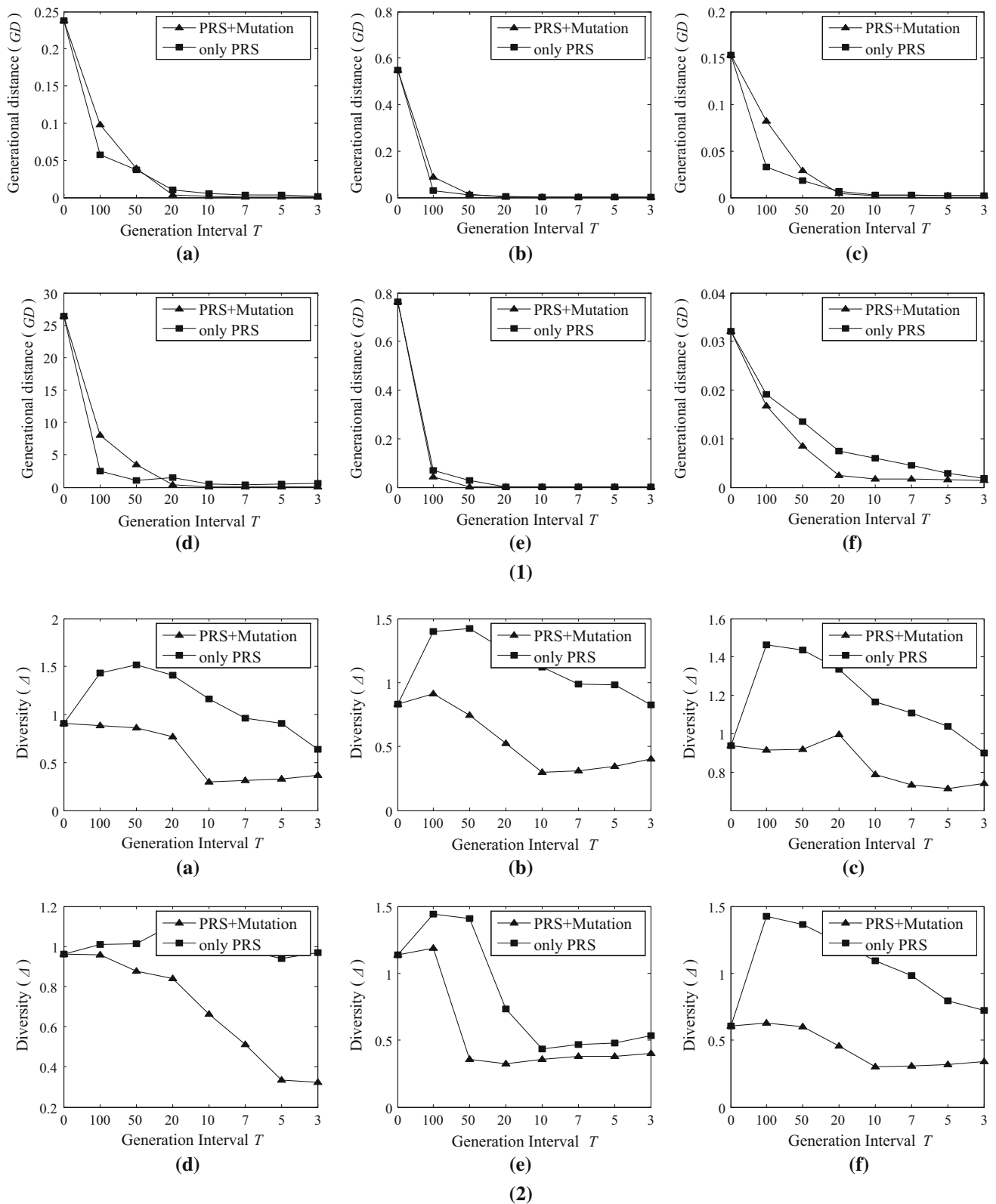


Fig. 5 The comparison between the RMOPSO (denote as PRS + Mutation) and the RMOPSO deleted the mutation operation (denote as only PRS). **1** Comparison of the convergence performance. Where **a** is ZDT1 problem, **b** is ZDT2, **c** is ZDT3, **d** is ZDT4, **e** is ZDT6, **f** is WFG1. **(2)** Comparison of the diversity performance. Where **a** is ZDT1 problem, **b** is ZDT2, **c** is ZDT3, **d** is ZDT4, **e** is ZDT6, **f** is WFG1

generate new population, so the new particle has higher probability to approach the PF compared to the original particles.

2. The introduction of new mutation strategy can improve the diversity of population, and then strengthen the ability to jump out of local optimum.
3. The combination of PRS and mutation strategy can improve both the convergence and diversity performance simultaneously.

3.5 Conclusion of the experimental analysis

In this section, through the experimental study on the influence of T for RMOPSO, it can be concluded that the convergence and diversity performance can be improved at the same time with decreasing T . The impacts of the PRS and mutation operators for the algorithm are investigated, the experimental results show that the case using only PRS can effectively strengthen the convergence performance, but it has no obvious advantage in diversity performance. The introduction of the mutation operators enhances the diversity of the population and then strengthens the ability to jump out of local optimum. The experimental results also verify that the combination of two proposed strategies always has better performance in both convergence and diversity.

4 Comparison with state-of-the-art algorithms

This section is devoted to present the comparison between proposed algorithms and state-of-the-art algorithms. The brief introduction and corresponding parameter setting for some state-of-the-art algorithms are given first. Next, the benchmark and performance indicator are used to evaluate the convergence and diversity performance. Finally, the results obtained by different algorithms are compared.

4.1 Algorithm and parameters settings

In this paper, NSGA-II, generalized differential evolution 3 (GDE3) (Kukkonen and Lampinen 2009), MOCLPSO, SMPSO, MOEA/D-DE (Li and Zhang 2009), CMPSO, 6 MOEAs and MOPSOs are selected to compare with RMOPSO. NSGA-II is the classical multi-objective evolutionary algorithm based on the Pareto dominate relation. GDE3 is a MOEA based on differential evolution (DE), which has a competitive performance. MOCLPSO is a multi-objective version of the CLPSO and famous for its ability of avoiding the local optimum. SMPSO introduces a velocity constriction mechanism and extends the range of parameters c_i for OMOPSO. It had been found to exhibit much better performance compared with some of the MOPSOs (Durillo et al. 2009). MOEA/D-DE is the DE version of MOEA/D which

Table 3 Parameters setting for all algorithms

Algorithms	Parameters setting
NSGA-II	$N = 100, p_c = 0.9, p_m = 1/n, \eta_c = 20$ and $\eta_m = 20$
GDE3	$N = 100, CR = 0.9$ and $F = 0.5$
SMPSO	$N = 100, w = \text{rand}(0.1, 0.5)$ $c_1 = \text{rand}(1.5, 2.5)$ and $c_2 = \text{rand}(1.5, 2.5)$, $p_m = 1/n$
MOCLPSO	$N = 50, p_c = 0.1, p_m = 0.4, w = 0.9 \rightarrow 0.2$ and $c = 2.0$
MOEA/D-DE	$N=100, CR=1.0, F=0.5, \eta = 20, p_m = 1/n, T=20, \delta = 0.9, n_r=2$
CMPSO	$N = 20, w = 0.9 \rightarrow 0.4, c_1 = c_2 = c_3 = 4.0/3$
RMOPSO	$N = 50, w = 0.729, c_1 = c_2 = 2.05, T = 5,$ $p_1 = 0.05, p_2 = 0.5$

employs SBX crossover operator as DE operator, CMPSO has competitive performance based on MPMP and ELS compared with some MOPSOs. Therefore, these algorithms are representative and helpful to make the comparisons more comprehensive and convincing.

The related parameters for all of algorithms are set according to the original literatures, as given in Table 3. In the experiments, each test problems will be independently executed for 30 times for each algorithm. Regarding different properties of problems, the maximum FEs is 25,000 for ZDT problems, 110^5 for DTLZ problems, 3×10^5 for UF problems. Except RMOPSO and CMPSO, the population size of algorithms is set to 100 for ZDT problems, 200 for DTLZ problems, 300 for UF problems, respectively. While the population size is 50 for RMOPSO, and 20 for CMPSO. The results obtained by different algorithms are compared with that by RMOPSO using the Wilcoxon rank sum test with significant level $\alpha = 0.05$.

4.2 Test problems

To verify the ability of the algorithm to deal with various problems, three-objective problems DTLZ1, DTLZ2 and DTLZ3 are added in this section, as they are complex problems with local optimum. The UF (Zhang et al. 2009) test set (UF1-7) with complicated Pareto sets is adopted here as well.

4.3 Performance metric

In this section, inverted generational distance (IGD) indicator is adopted, because it can measure the convergence and diversity performance simultaneously and had been widely used in the MOP community. The main difference between IGD and GD indicators is the changed order of the reference

set. *IGD* utilizes the Pareto set obtained by the algorithm as reference and then calculates the minimum distance away from the reference point that is generated before running to the Pareto set. If the Pareto set is not widely distributed, the distance will be large for the some reference point resulting in large *IGD* value. Therefore, only when the solutions are close to the *PF* and widely distributed, small *IGD* value can be obtained. The detailed description of the *IGD* indicator had been given in the Appendix 1.

4.4 Experimental results

The experimental results for the *IGD* indicator are given in Table 4. Table 5 reveals that the proposed RMOPSO obtains the best values in 7 out of the 15 test MOPs. NSGA-II and GDE3 have a competitive result for 1 problem. While MOEA/D-DE and CMPSO achieve competitive results for 3 problems. Moreover, for all of the ZDT series of MOPs, RMOPSO obtains the best result except the ZDT1 and ZDT2 cases, but the values obtained for these two cases are comparable with the best results obtained by CMPSO. For the three-objective DTLZ series of MOPs, RMOPSO obtains best result in 2 out of 3 problems, which are both complex problems. For DTLZ3 problem, the result obtained by RMOPSO is much better than others. For DTLZ2 problem, RMOPSO does not obtain the best result, but the result is still close to the best values obtained by CMPSO. In addition, RMOPSO has the best value of standard deviation (Std.), indicating a good robustness.

The ZDT4, DTLZ1, DTLZ3 are the complex problems used in the experiment to verify the algorithm's ability to jump out of local optimum. For the ZDT4 problem, RMOPSO has a much better performance compared with others, with a value of 0.0425. It should be noticed that this value is the mean of all the 30 independent runs, but the best value is 0.0036. The best value obtained by other algorithm is about 0.05. As can be seen from the above results, the RMOPSO exhibits best ability to jump out of local optimum for ZDT4 problem. For the DTLZ1 problems, the result for MOCLPSO is 18.75 and that for SMPSO is 0.65, and the CMPSO is 0.0567, all of these three algorithms are based on particle swarm optimization algorithm. Therefore, RMOPSO obtains much better result compared with MOPSOs (SMPSO and MOCLPSO, CMPSO). For the DTLZ3, all of the algorithms do not get convergence results but the result obtained by RMOPSO is found to be close to the *PF*.

From above results, the RMOPSO has been found to have stronger ability to overcome premature convergence problem and exhibits the best performance among the existing MOPSOs.

For UF series problems, RMOPSO obtains best value for UF2 and UF5, and MOEA/D-DE performs best on UF1, UF3 and UF7. CMPSO performs best on UF4 problem,

and NSGAI performs best on UF6 problem. For MOPSOs, RMOPSO performs best on UF2 and UF5 problems and has similar value on UF4 compared with the best one obtained by CMPSO. For other problems, it is obvious that MOEA/D-DE obtains best performance. Except MOEA/D-DE, other algorithms utilize the Pareto dominate relation to judge the merits of the solutions. The purpose to use Pareto-domination-based selection is to drive the whole population towards the PS or PF. However, it has no direct control on the movement of each individual in its population. No mechanism is employed to control the distribution of its computational efforts for different regions of the PF or PS. In contrast, MOEA/D decomposes MOPs into a set of single-objective sub-problems, the computational effort can be evenly distributed among these sub-problems. Therefore, MOEA/D-DE is found to outperform on UF series problems which have complicated PS.

Since none of algorithms can outperform all others on all the problems, the ranks of each algorithm on different problems are summed up to evaluate the overall performance, as shown in Table 4. Based on this definition, it can be found that the proposed RMOPSO ranks the first of all algorithms.

4.5 Impact of parameter settings

In Sect. 4.1, the parameter settings for RMOPSO are: $w = 0.729$, $c_1 = 2.05$ and $c_2 = 2.05$. To investigate the impact of w and c_i , the performance of RMOPSO with different values of w and c_i is studied. The study is conducted on unimodal objective functions (ZDT1 and DTLZ2) and multimodal objective functions (ZDT4 and DTLZ1).

The parameter settings used in SMPSO and MOCLPSO are adopted for comparison. The detailed information is listed as following:

Parameter 1: $w = 0.729$, $c_1 = 2.05$ and $c_2 = 2.05$.

Parameter 2: $w = \text{random}(0.1,0.5)$, $c_1 = \text{random}(1.5,2.5)$ and $c_2 = \text{random}(1.5,2.5)$.

Parameter 3: w is linearly decreasing from 0.9 to 0.2, and $c_1 = c_2 = 2.05$.

The two new parameter settings (parameter 2 and 3) are applied in RMOPSO, respectively. The obtained results are compared with those for the original parameters settings (parameter 1). The *IGD* indicator will be employed to evaluate the convergence property.

Table 5 shows the results obtained by different parameter settings. It reveals that parameter 2 exhibits better performance on RMOPSO to solve the complex problems (e.g. ZDT4 and DTLZ1). This is because the RMOPSO utilizes the recombination of every T generations and population distribution changes after recombination. The impact of the linear change of parameters is not obvious. So the random variation used by the SMPSO can achieve better results.

Table 4 IGD performance of all algorithms on different problems

	NSGAI	GDE3	MOEA/D-DE	SMP	CMPSO	MOCLPSO	RMOPSO
ZDT1							
MeanStd	0.0051 _{2.33E-04}	0.0127 _{1.56E-03}	0.0056 _{1.93E-04}	0.0071 _{7.83E-04}	0.0041 _{8.30E-05}	0.0048 _{1.76E-04}	0.0049 _{7.06E-08}
Rank	4 \approx	7-	6-	5-	1+	2 \approx	3
ZDT2							
MeanStd	0.009 _{2.80E+00}	0.0029 _{1.82E-02}	0.0053 _{3.07E-05}	0.0061 _{3.81E-04}	0.0043 _{1.03E-04}	0.0038 _{3.11E-04}	0.0046 _{6.64E-08}
Rank	7-	1+	5-	6-	3 \approx	2+	4
ZT3							
MeanStd	0.0075 _{2.71E-02}	0.0116 _{2.24E-03}	0.0055 _{2.49E-04}	0.023 _{5.99E-03}	0.0139 _{3.49E-03}	0.055 _{2.49E-04}	0.0016 _{1.01E-04}
Rank	3-	4-	2-	6-	5-	7-	1
ZDT4							
MeanStd	0.29 _{4.00E-01}	0.34 _{3.70E-01}	0.31 _{0.23}	0.56 _{4.12E-01}	0.79 _{0.26}	3.26 _{1.35E+00}	0.0425 _{1.25E-02}
Rank	2-	4-	3-	5-	6-	7-	1
ZDT6							
MeanStd	0.0062 _{7.02E-04}	0.0736 _{9.16E-02}	0.0054 _{0.13E-04}	0.0046 _{3.36E-04}	0.0037 _{1.47E-04}	0.0087 _{1.31E-04}	0.0028 _{5.91E-08}
Rank	5-	7-	4-	3-	2 \approx	6-	1
DTLZ1							
MeanStd	0.0042 _{2.86E-04}	0.0024 _{1.34E-04}	0.0028 _{4.36E-04}	0.65 _{8.06E-01}	0.0567 _{2.21E-02}	18.75 _{7.36E+00}	0.0021 _{5.79E-04}
Rank	4-	2 \approx	3 \approx	6-	5-	7-	1
DTLZ2							
MeanStd	0.0058 _{4.70E-04}	0.0055 _{6.85E-04}	0.0333 _{3.02E-03}	0.0067 _{6.08E-04}	0.0046 _{1.50E-04}	0.0088 _{8.06E-04}	0.0056 _{3.98E-05}
Rank	4-	2 \approx	7-	5-	1+	6-	3
DTLZ3							
MeanStd	0.8271 _{6.50E-01}	0.7216 _{5.90E-01}	0.5613 _{6.32E-03}	0.96 _{5.32E-01}	0.6268 _{2.95E-03}	12.35 _{1.62E+01}	0.0695 _{1.90E-01}
Rank	5-	4-	2-	6-	3-	7-	1
UF1							
MeanStd	0.073 _{2.46E-02}	0.0595 _{2.48E-02}	0.0576 _{2.15E-02}	0.0981 _{7.91E-03}	0.0664 _{1.99E-02}	0.10 _{7.17E-03}	0.1022 _{1.15E-04}
Rank	4+	2+	1+	5+	3+	6 \approx	7
UF2							
MeanStd	0.0206 _{3.67E-03}	0.0202 _{3.81E-03}	0.0663 _{1.32E-02}	0.0724 _{3.54E-03}	0.0169 _{3.37E-03}	0.11 _{3.39E-03}	0.0119 _{2.81E-06}
Rank	4-	3-	5-	6-	2-	7-	1
UF3							
MeanStd	0.0695 _{1.14E-02}	0.16 _{6.66E-02}	0.0389 _{1.57E-02}	0.37 _{9.71E-03}	0.098 _{1.39E-02}	0.48 _{1.55E-02}	0.3704 _{1.49E-04}
Rank	2+	4+	1+	5 \approx	3+	7-	6
UF4							
MeanStd	0.0426 _{4.46E-04}	0.0472 _{1.03E-03}	0.0295 _{1.59E-03}	0.16 _{1.39E-02}	0.0238 _{1.90E-03}	0.12 _{1.10E-02}	0.0338 _{2.59E-06}
Rank	4-	5-	2+	7-	1+	6-	3
UF5							
MeanStd	0.32 _{8.41E-02}	0.21 _{1.61E-02}	0.33 _{5.41E-02}	0.74 _{0.12}	0.20 _{2.01E-02}	0.51 _{0.18}	0.1941 _{1.60E-04}
Rank	4-	3-	5-	7-	2-	6-	1
UF6							
MeanStd	0.12 _{1.93E-02}	0.30 _{1.72E-02}	0.14 _{9.05E-02}	0.40 _{3.40E-02}	0.1491 _{2.04E-02}	0.4046 _{4.30E-02}	0.2841 _{1.89E-02}
Rank	1+	5-	2+	6-	3+	7-	4
UF7							
MeanStd	0.16 _{0.16}	0.0297 _{1.02E-03}	0.0083 _{9.40E-04}	0.22 _{0.15}	0.12 _{0.13}	0.19 _{0.15}	0.0459 _{2.68E-06}
Rank	5-	2+	1+	7-	4-	6-	3

Table 4 continued

	NSGAI	GDE3	MOEA/D-DE	SMP	CMPSO	MOCLPSO	RMOP
Sum of rank	58	55	49	86	44	89	40
Final rank	5	4	3	6	2	7	1
≈	1	2	1	1	2	2	
+	3	4	5	1	6	1	
–	11	9	9	13	7	12	

Wilcoxon's rank sum test at a 0.05 significance level is performed between RMOPSO and each of algorithms. "≈", "+" and "–" denote that the performance of the corresponding algorithm is similar, significantly better than or worse than that of RMOPSO, respectively. The row of "≈", "+" and "–" shows the number of "≈", "+" and "–" for each algorithm. The best mean is highlighted in boldface

Table 5 The *IGD* performance of RMOPSO with different parameter settings

	ZDT1	ZDT4	DTLZ1	DTLZ2
Parameter 1	0.0045 _{1.30E-05}	0.0569 _{2.47E-03}	0.0073 _{5.90E-05}	0.0036 _{2.21E-04}
Parameter 2	0.0045 _{2.35E-05}	0.0075 _{2.52E-03}	0.0046 _{3.09E-05}	0.0035 _{3.09E-04}
Parameter 3	0.0049 _{3.25E-05}	0.0091 _{1.36E-03}	0.0057 _{3.16E-05}	0.0035 _{1.87E-04}

4.6 Discussion

In the pseudo-code of the *PRS* given in Fig. 3, p_1 is used to determine how the mutation will be applied on corresponding dimension. If $p_1 = 1$, the *PRS* will be no any effect. If $p_1 = 0$, then each dimension will be replaced. To maximize the effectiveness of *PRS*, p_1 is set to 0.05. p_2 is used to determine the selection of *pbest* and *gbest*. If $p_2 = 1$, the *gbest* will not be selected; if $p_2 = 0$, the *pbest* will not be selected; if p_2 is set to 0.5, *pbest* and *gbest* have the same probability to be selected. Therefore, p_2 is used to balance the effects of the *pbest* and *gbest*. However, the accurate sets of p_1 and p_2 need further experimental studies. *PRS* is similar to the cross operation used in GA, although the comparison with some cross operations like SBX (only replaced *PRS* with SBX in RMOPSO) found that *PRS* outperforms SBX. The relevant investigation will be considered as the future work.

5 Conclusion

In this paper, the population recombination strategy is first proposed to avoid premature convergence occurred in MOPSOs. The *PRS* effectively uses all variable information found by MOPSO to construct new population and makes the new population inherit the original particle's position information. Since there is no new variable information created in the population recombination process, a new mutation strategy is introduced to strengthen diversity performance. The experiment to analyze the effects of the proposed mutation strategy is also carried out. Experimental results show that mutation operation can improve the diversity and convergence performance simultaneously.

The proposed algorithms are also compared with other state-of-the-art MOEAs and MOPSOs. According to the

results, for the simple test problems, the performance of the proposed algorithm and other algorithms is comparable. However, for the complex test problems, the performance of the proposed algorithm is better even superior to other algorithms. It has been also proved that the use of *PRS* and new mutation strategy strengthen the ability to jump out of local optimum.

Acknowledgements The work was supported in part by the National Natural Science Foundation of China (No. 61401523), in part by the Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (No. 2014KQNCX002), in part by the International Science & Technology Cooperation Program of China (No. 2015DFR11050), and in part by the External Cooperation Program of Guangdong Province of China (No. 2013B051000060).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1: Performance measures

The three performance indicators used in this paper are described as follows.

Generational distance (GD): The *GD* metric is defined as:

$$GD(A, P) = \frac{\sum_{i=1}^{|A|} d(A_i, P)}{|A|}$$

where A is the archive found by the algorithm and P is the reference point set (500 point in the experiment) in *PF* generated prior. d_i is the minimum distance between the A_i and the point in P . Obviously, this value can only measure the distance from the archive to the *PF*, and the distribution of the archive cannot be evaluated. Therefore, the Δ indicator is

employed for the comprehensive evaluation of the algorithm together with GD .

Diversity indicator: Δ measures the extreme and spread among the obtained solutions:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}$$

Here, d_i is Euclidean distance between nearby solutions in A and \bar{d} is the average of these distances. d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the A . N is the number of solutions in A .

Inverted generational distance (IGD): The IGD metric is defined as:

$$IGD(P, A) = \frac{\sum_{i=1}^{|P|} d(P_i, A)}{|P|}$$

Here the definition of P and A is the same as those for GD indicator, but d_i is the closest distance from the point in P to the point in A . It can be used to evaluate both convergence and diversity performance simultaneously.

References

- Coello C, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8(3):256–279
- Daneshyari M, Yen GG (2011) Cultural-based multiobjective particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern* 41(2):553–567
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester
- Deb K, Pratap A, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Durillo J, Nieto G, Nebro A, Coello C, Luna F, Alba E (2009) Multi-objective particle swarm optimizers: an experimental comparison. In: 5th international conference on evolutionary multi-criterion optimization (EMO'2009)
- Huang V, Suganthan P, Liang J (2006) Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *Int J Intell Syst* 21(2):209–226
- Huband S, Barone L, While L, Hingston P (2005) *A scalable multi-objective test problem toolkit*. Lecture notes in computer science. Springer, Berlin, Germany
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural network, Perth, Australia, vol 4, pp 1942–1948
- Kukkonen S, Lampinen J (2009) Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In: Proceedings of the IEEE congress on evolutionary computation, pp 1943–1950
- Lalwani S, Singhal S, Kumar R, Gupta N (2013) A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Trans Combin* 2(1):39–101
- Li X (2003) A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: Cant' u-Paz E et al (eds) GECCO'2003, pp 37–48
- Liang J, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302
- Luis V, Santana-Quintero, Noel R, Coello C (2006) A multi-objective particle swarm optimizer hybridized with scatter search. In: Gelbukh A, Reyes-Garcia CA (eds) MICAI 2006 LNAI 4293. Springer, Berlin, Heidelberg, pp 294–304
- Margarita R-S, Carlos A, Coello C (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int J Comput Intell Res* 2(3):287–308
- Moore J, Chapman R (1999) Application of particle swarm to multi-objective optimization. Department of Computer Science and Software Engineering, Auburn University, Auburn
- Mostaghim S, Teich J (2003) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: Proceedings of 2003 IEEE swarm intelligence symposium, Indianapolis, Indiana, USA, pp 26–33
- Moubayed NA, Petrovski A, McCall J (2010) A novel smart multi-objective particle swarm optimization using decomposition. In: Schaefer R et al (eds) PPSN XI, Part II, LNCS 6239, pp 1–10
- Nebro AJ, Durillo J, García-Nieto J, Coello C, Luna F, Alba E (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: Proceedings of the IEEE symposium on computational intelligence MCDM, pp 66–73
- Paul S. Andrews (2006) An investigation into mutation operators for particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation
- Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of the IEEE World congress computational intelligence, pp 69–73
- Sierra MR, Coello C (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance. *Lect Notes Comput Sci* 3410:505–519
- Tang L, Wang X (2013) A hybrid multi-objective evolutionary algorithm for multi-objective optimization problems. *IEEE Trans Evol Comput* 17(1):20–46
- Tate J, Benjamin W-L, Bate I, Yao X (2012) Evolutionary and principled search strategies for sensor network protocol optimization. *IEEE Trans Syst Man Cybern B Cybern* 42(1):163–180
- Xia H, Zhuang J, Yu D (2014) Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization. *IEEE Trans Cybern* 44(3):378–393
- Xue B, Zhang M, Browne WN (2013) Particle swarm optimization for feature selection in classification: a multi-objective approach. *IEEE Trans Cybern* 43(6):1656–1671
- Zhan ZH, Zhang J, Li Y, Shu-Hung C (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern* 39(6):1362–1381
- Zhang Q, Li H (2007) MOEA/D: a multi-objective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Zhang Q, Zhou A, Zhao S, Suganthan P, Liu W, Tiwari S (2009) Multi-objective optimization test instances for the CEC 2009 special session and competition. In: Proceedings of the IEEE congress on evolutionary computation, pp 1–30
- Zhan Z, Zhang J et al (2013) Multiple populations for multiple objectives: a co-evolutionary technique for solving multi-objective optimization problems. *IEEE Trans Cybern* 43(2):445–463