

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A Gradient Multiobjective Particle Swarm Optimization

Hong-Gui Han, Lu Zhang and Jun-Fei Qiao

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.76306>

Abstract

An adaptive gradient multiobjective particle swarm optimization (AGMOPSO) algorithm, based on a multiobjective gradient (MOG) method, is developed to improve the computation performance. In this AGMOPSO algorithm, the MOG method is devised to update the archive to improve the convergence speed and the local exploitation in the evolutionary process. Attributed to the MOG method, this AGMOPSO algorithm not only has faster convergence speed and higher accuracy but also its solutions have better diversity. Additionally, the convergence is discussed to confirm the prerequisite of any successful application of AGMOPSO. Finally, with regard to the computation performance, the proposed AGMOPSO algorithm is compared with some other multiobjective particle swarm optimization (MOPSO) algorithms and two state-of-the-art multiobjective algorithms. The results demonstrate that the proposed AGMOPSO algorithm can find better spread of solutions and have faster convergence to the true Pareto-optimal front.

Keywords: multiobjective particle swarm optimization, multiobjective problem, multiobjective gradient, convergence

1. Introduction

Most of the engineering and practical applications, such as wastewater treatment processes and aerodynamic design problem, often have a multiobjective nature and require solving several conflicting objectives [1–3]. Handling with these multiobjective optimization problems (MOPs), there are always a set of possible solutions which represent the tradeoffs among the objectives known as Pareto-optimal set [4–5]. Evolutionary multiobjective optimization (EMO) algorithms, which are a class of stochastic optimization approaches based on population characteristic, are widely used to solve the MOPs, because a series of Pareto-optimal solutions can be obtained in a single run [6–9]. The multiobjective optimization algorithms are striving to acquire a Pareto-optimal set with good diversity and convergence. The most typical EMO algorithms include the

non-dominated sorting genetic algorithm (NSGA) [10] and NSGA-II [11], the strength Pareto evolutionary algorithm (SPEA) [12] and SPEA2 [13], the Pareto archived evolutionary strategy (PAES) [14], the Pareto envelope-based selection algorithm (PESA) [15] and PESA-II [16].

The notable characteristic of particle swarm optimization (PSO) is the cooperation among all particles of a swarm which is attracted toward the global best (gBest) in the swarm and its own personal best (pBest), so the PSO have a better global searching ability [17–19]. Among these EMO algorithms, owing to the high convergence speed and ease of implementation, multiobjective particle swarm optimization (MOPSO) algorithms have been widely used [20–23]. MOPSO can also be applied to multiple difficult optimization problems such as noisy and dynamic problems. However, apart from the archive maintenance, in MOPSO, two issues still remain to be further addressed. The first one is the update of gBest and pBest, because the absolute best solution cannot be selected by the relationship of the non-dominated solutions. Then, the selection of gBest and pBest results in the different flight directions for a particle, which has an important effect on the convergence and diversity of MOPSO [24].

Zheng et al. introduced a novel MOPSO algorithm, which can improve the diversity of the swarm and improve the performance of the evolving particles over some advanced MOPSO algorithms with a comprehensive learning strategy [25]. The experimental results illustrate that the proposed approach performs better than some existing methods on the real-world fire evacuation dataset. In [26], a multiobjective particle swarm optimization with preference-based sort (MOPSO-PS), in which the user's preference was incorporated into the evolutionary process to determine the relative merits of non-dominated solutions, was developed to choose the suitable gBest and pBest. The results indicate that the user's preference is properly reflected in optimized solutions without any loss of overall solution quality or diversity. Moubayed et al. proposed a MOPSO by incorporating dominance with decomposition (D2MOPSO), which proposes a novel archiving technique that can balance the relationship of the diversity and convergence [27]. The analysis of the comparable experiments demonstrates that the D2MOPSO can handle with a wide range of MOPs efficiently. And some other methods for the update of gBest and pBest can be found in [28–31]. Although many researches have been done, it is still a huge challenge to select the appropriate gBest and pBest with the suitable convergence and diversity [32–33].

The second particular issue of MOPSO is how to own fast convergence speed to the Pareto Front, well known as one of the most typical features of PSO. According to the requirement of the fast convergence for MOPSO, many different strategies have been put forward. In [34], Hu et al. proposed an adaptive parallel cell coordinate system (PCCS) for MOPSO. This PCCS is able to select the gBest solutions and adjust the flight parameters based on the measurements of parallel cell distance, potential and distribution entropy to accelerate the convergence of MOPSO by assessing the evolutionary environment. The comparative results show that the self-adaptive MOPSO is better than the other methods. Li et al. proposed a dynamic MOPSO, in which the number of swarms is adaptively adjusted throughout the search process [35]. The dynamic MOPSO algorithm allocates an appropriate number of swarms to support convergence and diversity criteria. The results show that the performance of the proposed dynamic MOPSO algorithm is competitive in comparison to the selected algorithms on some standard benchmark problems. Daneshyari et al. introduced a cultural framework to design a flight

parameter mechanism for updating the personalized flight parameters of the mutated particles in [36]. The results show that this flight parameter mechanism performs efficiently in exploring solutions close to the true Pareto front. In the above MOPSO algorithms, the improved strategies are expected to achieve better performance. However, few works have been done to examine the convergence of these MOPSO algorithms [37].

Motivated by the above review and analysis, in this chapter, an adaptive gradient multiobjective particle swarm optimization (AGMOPSO) algorithm, based on a multiobjective gradient (MOG) method, is put forward. This novel AGMOPSO algorithm has faster convergence in the evolutionary process and higher efficiency to deal with MOPs. The proposed AGMOPSO algorithm contains a major contribution to solve MOPs as follows: A novel MOG method is proposed for updating the archive to improve the convergence speed and the local exploitation in the evolutionary process. Unlike some existing gradient methods for single-objective optimization problems [38–40] and MOPs [41], much less is known about the gradient information of MOPs. One of the key features of the MOG strategy is that the utilization of gradient information for MOPs is able to obtain a Pareto set of solutions to approximate the optimal Pareto set. In view of the advantages of the MOG strategy, this AGMOPSO algorithm can obtain a good Pareto set and reach smaller testing error with much faster speed. This characteristic makes this method ideal for MOPs.

2. Problem formulation

2.1. Multiobjective problems

A minimize MOP contains several conflicting objectives which is defined as:

$$\begin{aligned} &\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \\ &\text{subject to } \mathbf{x} \in \Omega, \end{aligned} \quad (1)$$

where m is the number of objectives, \mathbf{x} is the decision variable, $f_i()$ is the i th objective function. A decision variable \mathbf{y} is said to dominate the decision vector \mathbf{z} , defined as \mathbf{y} dominates \mathbf{z} or $\mathbf{y} < \mathbf{z}$, which is indicated as:

$$\forall i : f_i(\mathbf{y}) \leq f_i(\mathbf{z}) \text{ and } \exists j : f_j(\mathbf{y}) < f_j(\mathbf{z}), \quad (2)$$

where $i = 1, 2, \dots, m, j = 1, 2, \dots, m$. When there is no solution that can dominate one solution in MOPs, this solution can be used as the Pareto optimal solution. This Pareto optimal solution comprises the Pareto front.

2.2. Particle swarm optimization

PSO is a stochastic optimization algorithm, in which a swarm contains a certain number of particles that the position of each particle can stand for one solution. The position of a particle which is expressed by a vector:

$$\mathbf{x}_i(t) = [x_{i,1}(t), x_{i,2}(t), \dots, x_{i,D}(t)], \quad (3)$$

where D is the dimensionality of the searching space, $i = 1, 2, \dots, s$; s is the swarm size. Also each particle has a velocity which is represented as:

$$\mathbf{v}_i(t) = [v_{i,1}(t), v_{i,2}(t), \dots, v_{i,D}(t)]. \quad (4)$$

During the movement, the best previous position of each particle is recorded as $\mathbf{p}_i(t) = [p_{i,1}(t), p_{i,2}(t), \dots, p_{i,D}(t)]$, and the best position obtained by the swarm is denoted as $\mathbf{g}(t) = [g_1(t), g_2(t), \dots, g_D(t)]$. Based on $\mathbf{p}_i(t)$ and $\mathbf{g}(t)$, the new velocity of each particle is updated by:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 (p_{i,d}(t) - x_{i,d}(t)) + c_2 r_2 (g_d(t) - x_{i,d}(t)), \quad (5)$$

where t denotes the t th iteration during the searching process; $d = 1, 2, \dots, D$ is the dimension in the searching space; ω is the inertia weight; c_1 and c_2 are the acceleration constants and r_1 and r_2 are the random values uniformly distributed in $[0, 1]$. Then the updating formula of the new position is expressed as:

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1). \quad (6)$$

At the beginning of the searching process, the initial position of each particle is randomly generated. As the searching process goes on, the particle swarm may appear as an uneven distribution phenomenon in the evolutionary space.

3. Multiobjective gradient method

The key points of AGMOPSO, compared to the original MOPSO, are that the MOG method is taken into account. In AGMOPSO, the population with N particles intends to search for a set of non-dominated solutions to be stored in an archive with a predefined maximal size.

In MOPSO, the position of each particle can represent the potential solution for the conflicting objectives. The gBest and pBest can guide the evolutionary direction of the whole particle swarm. The position \mathbf{x}_i and velocity \mathbf{v}_i of the i th particle are the D -dimensional vectors $\mathbf{x}_i(0) \in \mathbb{R}_D$, $\mathbf{v}_i(0) \in \mathbb{R}_D$. The particle updates the velocity and position by the motion trajectory in Eqs. (5) and (6). The external archive $\mathbf{A}(0)$ is initialized as a null set. Meanwhile, the best previous position $\mathbf{p}_i(t)$ is computed by:

$$\mathbf{p}_i(t) = \begin{cases} \mathbf{p}_i(t-1), & \text{if } \mathbf{x}_i(t) < \mathbf{p}_i(t-1), \\ \mathbf{x}_i(t), & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbf{a}_j(t-1) < \mathbf{p}_i(t)$ means $\mathbf{x}(t)$ is not dominated by $\mathbf{p}_i(t-1)$. The process of archive $\mathbf{A}(t)$ is updated based on the previous archive $\mathbf{A}(t-1)$ and the best previous position $\mathbf{p}_i(t)$

$$\mathbf{A}(t) = \begin{cases} \mathbf{A}(t-1) \cup \mathbf{p}_i(t), & \text{if } \mathbf{a}_j(t-1) < > \mathbf{p}_i(t), \\ \bar{\mathbf{A}}(t-1) \cup \mathbf{p}_i(t), & \text{otherwise,} \end{cases} \quad (8)$$

where $\mathbf{A}(t) = [\mathbf{a}_1(t), \mathbf{a}_2(t), \dots, \mathbf{a}_K(t)]^T$, $\mathbf{a}_j(t) = [a_{1,j}(t), a_{2,j}(t), \dots, a_{D,j}(t)]$, $\bar{\mathbf{A}}(t-1)$ is updated archive which removed the solutions dominated by the best previous position $\mathbf{p}_i(t)$, K is the dimensionality of archive $\mathbf{A}(t)$ which will be changed in the learning process, $\mathbf{a}_j(t-1) < > \mathbf{p}_i(t)$ means $\mathbf{a}_j(t-1)$ is not dominated by $\mathbf{p}_i(t)$ and $\mathbf{p}_i(t)$ is not dominated by $\mathbf{a}_j(t-1)$. Moreover, $\mathbf{g}(t)$ is found according to [24].

In AGMOPSO, to enhance the local exploitation, the archive $\mathbf{A}(t)$ is further updated by the MOG method using the gradient information to obtain a Pareto set of solutions that approximates the optimal Pareto set. Without loss of generality, assuming all of the objective functions are differentiable, the directional derivative in $f_i(\mathbf{a}_j(t))$ in a direction $\bar{\mathbf{u}}_j(t)$ at point $\mathbf{a}_j(t)$ is denoted as

$$\nabla_{\bar{\mathbf{u}}_j(t)} f_i(\mathbf{a}_j(t)) = \lim_{\delta \rightarrow 0} \left\{ \frac{f_i(\mathbf{a}_j(t) + \delta \bar{\mathbf{u}}_j(t)) - f_i(\mathbf{a}_j(t))}{\delta} \right\}, \quad (9)$$

where $\delta > 0$, $\bar{\mathbf{u}}_j(t) = [\bar{u}_{1,j}(t), \bar{u}_{2,j}(t), \dots, \bar{u}_{D,j}(t)]$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, K$, and the directional derivative can be rewritten:

$$\nabla_{\bar{\mathbf{u}}_j(t)} f_i(\mathbf{a}_j(t)) = \nabla f_i(\mathbf{a}_j(t)) \bar{\mathbf{u}}_j(t), \quad (10)$$

then, the gradient direction of MOP can be represented as:

$$\nabla_{\bar{\mathbf{u}}_j(t)} \mathbf{F}(\mathbf{a}_j(t)) = \left[\nabla_{\bar{\mathbf{u}}_j(t)} f_1(\mathbf{a}_j(t)), \nabla_{\bar{\mathbf{u}}_j(t)} f_2(\mathbf{a}_j(t)), \dots, \nabla_{\bar{\mathbf{u}}_j(t)} f_m(\mathbf{a}_j(t)) \right]^T, \quad (11)$$

According to Eq. (11), the minimum direction of MOP is calculated as

$$\hat{\mathbf{u}}_i(t) = \frac{\nabla f_i(\mathbf{a}_j(t))}{\|\nabla f_i(\mathbf{a}_j(t))\|},$$

$$\nabla f_i(\mathbf{a}_j(t)) = [\partial f_i(\mathbf{a}_j(t)) / \partial a_{1,j}(t), \partial f_i(\mathbf{a}_j(t)) / \partial a_{2,j}(t), \dots, \partial f_i(\mathbf{a}_j(t)) / \partial a_{D,j}(t)], \quad (12)$$

and $\|\hat{\mathbf{u}}_i(t)\| = 1$. In addition, the smooth criteria $f_i(\mathbf{a}_j(t))$ are said to be Pareto-stationary at the point $\mathbf{a}_j(t)$ if

$$\sum_{i=1}^m \alpha_i(t) \hat{\mathbf{u}}_i(t) = \mathbf{0}, \quad \sum_{i=1}^m \alpha_i(t) = 1, \quad \alpha_i(t) \geq 0, \quad (\forall i). \quad (13)$$

The weight vector can be set as

$$\boldsymbol{\alpha}(t) = \frac{1}{\|\hat{\mathbf{U}}^T \hat{\mathbf{U}}\|^2} \left[\|\hat{\mathbf{u}}_1\|^2, \|\hat{\mathbf{u}}_2\|^2, \dots, \|\hat{\mathbf{u}}_m\|^2 \right]^T, \quad (14)$$

where $\hat{\mathbf{U}}(t) = [\hat{\mathbf{u}}_1(t), \hat{\mathbf{u}}_2(t), \dots, \hat{\mathbf{u}}_m(t)]$, $\alpha_i(t) = \|\hat{\mathbf{u}}_i\|^2 / \|\hat{\mathbf{U}}^T \hat{\mathbf{U}}\|^2$, and $\|\boldsymbol{\alpha}\| = 1$.

To find the set of Pareto-optimal solutions of MOPs, the multi-gradient descent direction is given as follows:

$$\nabla F(\mathbf{a}_j(t)) = \sum_{i=1}^m \alpha_i(t) \hat{\mathbf{u}}_i(t), \quad \sum_{i=1}^m \alpha_i(t) = 1, \quad \alpha_i(t) \geq 0, \quad (\forall i). \quad (15)$$

This multi-gradient descent direction is utilized to evaluate the full set of unit directions. And the archive $\mathbf{A}(t)$ is updated as follows:

$$\bar{\mathbf{a}}_j(t) = \mathbf{a}_j(t) + h \cdot \nabla F(\mathbf{a}_j(t)), \quad (16)$$

where, h is the step size, $\mathbf{a}_j(t)$ and $\bar{\mathbf{a}}_j(t)$ are the j th archive variables before and after the MOG algorithm has been used at time t and the fitness values are updated at the same time.

Moreover, the archive $\mathbf{A}(t)$ can store the non-dominated solutions of AGMOPSO. But the number of non-dominated solutions will gradually increase during the search process. Therefore, to improve the diversity of the solutions, a fixed size archive is implemented in AGMOPSO to record the good particles (non-dominated solutions). During each iteration, the new solutions will be compared with the existing solutions in the archive using the dominating relationship. When a new solution cannot be dominated by the existing solutions in the archive, it will be reserved in the archive. On the contrary, the dominated new solutions cannot be accepted in the archive. If the capacity of the archive reaches the limitation, a novel pruning strategy is proposed to delete the redundant non-dominated solutions to maintain uniform distribution among the archive members.

Assuming that there are K points which will be selected from the archive serve. The maximum distance of the line segment between the first and the end points (namely whole Euclidean distance D_{max}) are obtained. Then, the average distance of the remained $K-2$ points are set

$$d = D_{max} / (K - 1), \quad (17)$$

where d is the average distance of all points. The average values of d are used to guide to select the non-dominated solutions of more uniform distribution. In addition, for the three objectives, all of the solutions (except the first and the end) are projected to the D_{max} . The points can be reserved, the projective points and the average distance points can be found. However, most projective distances of the adjacent points are not equal to the average distance. Thus, the next point is likely to be selected when it has the distance more closely to the average distance. Once the search process is terminated, the solutions in archive will become the final Pareto front. Taking DTLZ2 as an example, **Figure 1** shows this strategy with three objectives in details.

Local search is a heuristic method to improve PSO performance. It repeatedly tries to improve the current solution by replacing it with a neighborhood solution. In the proposed MOG algorithm, the set of unit directions is described by the normalized combination of the unit directions that map to the intersection points as Eq. (12). Then, each single run of the algorithm can yield a set of Pareto solutions. Experiments demonstrate that the improvements make AGMOPSO effective.

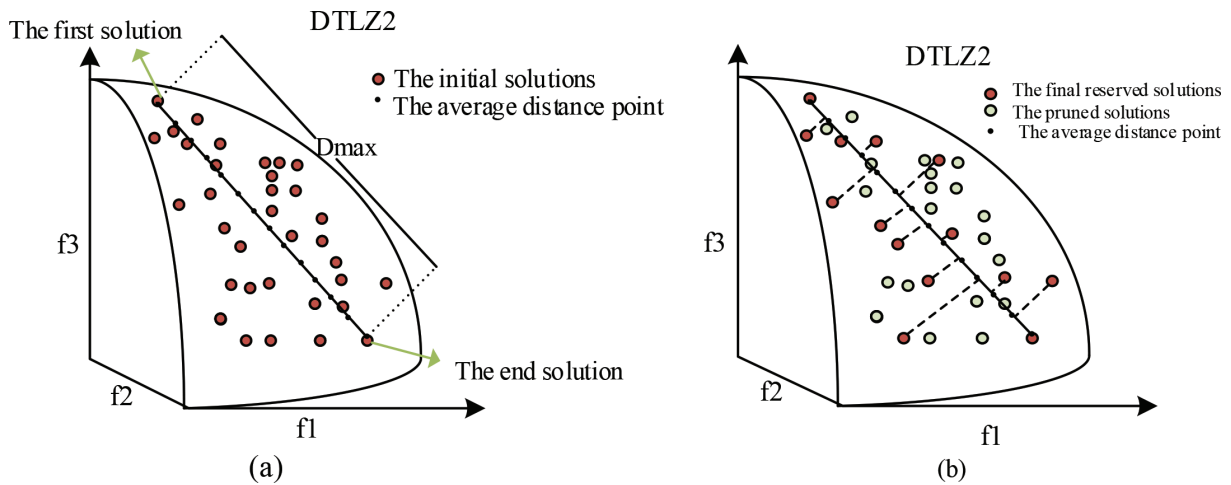


Figure 1. Illustration of points selection procedure. (a) Is the original points and (b) is the selection result of the proposed strategy.

Initializing the flight parameters, population size, the particles positions $x(0)$ and velocity $v(0)$

Loop

Calculating the fitness value

Getting the non-dominated solutions

% Eq. (8)

Storing the non-dominated solutions in archive $A(t)$

Updating the archive using MOG method

% Eq. (16)

If (the number of archive solutions exceed capacity)

Pruning the archive

End

Selecting the gBest from the archive $A(t)$

Calculating the flight parameters

Updating the velocity $x_i(t)$ and position $v_i(t)$

% Eqs. (5–6)

End loop

Table 1. AMOPSO algorithm.

In MOPSO, it is desired that an algorithm maintains good spread of solutions in the non-dominated solutions as well as the convergence to the Pareto-optimal set. In this AGMOPSO algorithm, an estimate of density is designed to evaluate the density of solutions surrounding it. It calculates the overall Euclidean distance values of the solutions, and then the average distance of the solutions along each of the objectives corresponding to each objective is calculated. This method is able to get a good spread result under some situations to improve the searching ability. And the pseudocode of AGMOPSO is presented in **Table 1**.

4. Simulation results and analysis

In this section, three ZDT and two DTLZ benchmark functions are employed to test the proposed of AGMOPSO. This section compares the proposed AGMOPSO with four state-of-the-art MOPSO algorithms—adaptive gradient MOPSO (AMOPSO) [41], crowded distance MOPSO (cdMOPSO) [32], pdMOPSO [31] and NSGA-II [11].

4.1. Performance metrics

To demonstrate the performance of the proposed AGMOPSO algorithm, two different quantitative performance metrics are employed in the experimental study.

1. Inverted generational distance (*IGD*):

$$IGD(F^*, F) = \sum_{x \in F^*} \text{mindis}(x, F) / |F^*|, \quad (18)$$

where $\text{mindis}(x, F)$ is the minimum Euclidean distance between the solution x and the solutions in F . A smaller value of $IGD(F^*, F)$ demonstrates a better convergence and diversity to the Pareto-optimal front.

2. Spacing (*SP*):

$$SP = \sqrt{\frac{1}{K-1} \sum_{i=1}^K (\bar{d} - d_i)^2}, \quad (19)$$

where d_i is the minimum Euclidean distance between i th solution and other solutions, K is the number of non-dominated solutions, \bar{d} is the average distance of the all Euclidean distance d_i .

4.2. Parameter settings

All the algorithms have three common parameters: the population size N , the maximum number of non-dominated solutions K and iterations T . Here, $N = 100$, $K = 100$ and $T = 3000$.

4.3. Experimental results

The experimental performance comparisons of the cdMOPSO algorithm on ZDTs and DTLZs are shown in **Figures 2–6**. Seen from **Figures 2–6**, the non-dominated solutions obtained by the proposed AGMOPSO algorithm can approach to the Pareto Front appropriately and maintain

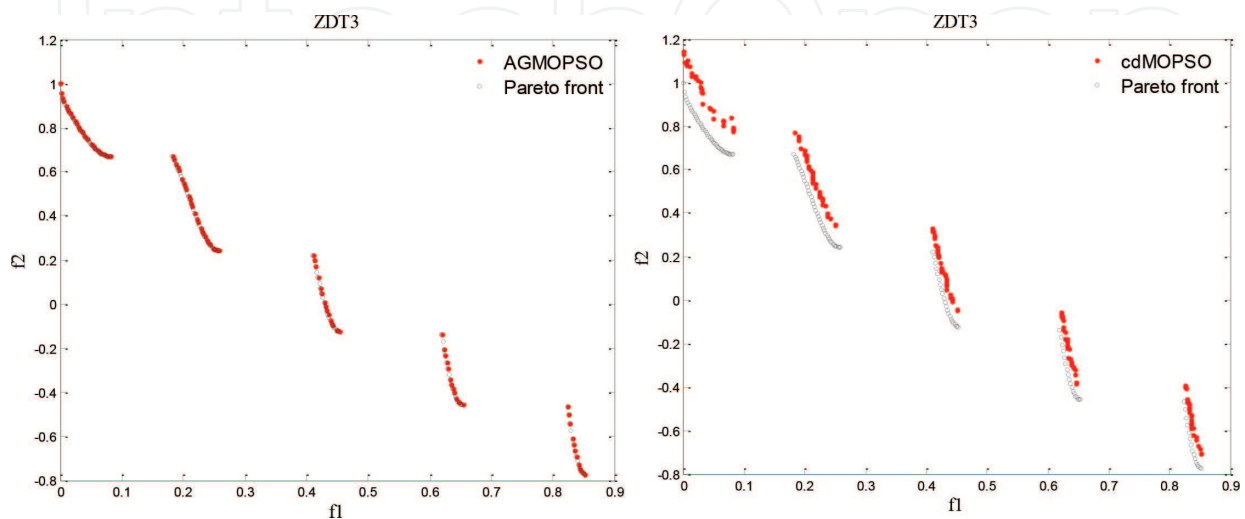


Figure 2. The Pareto front with non-dominated solutions obtained by the two multiobjective algorithms for ZDT3 function.

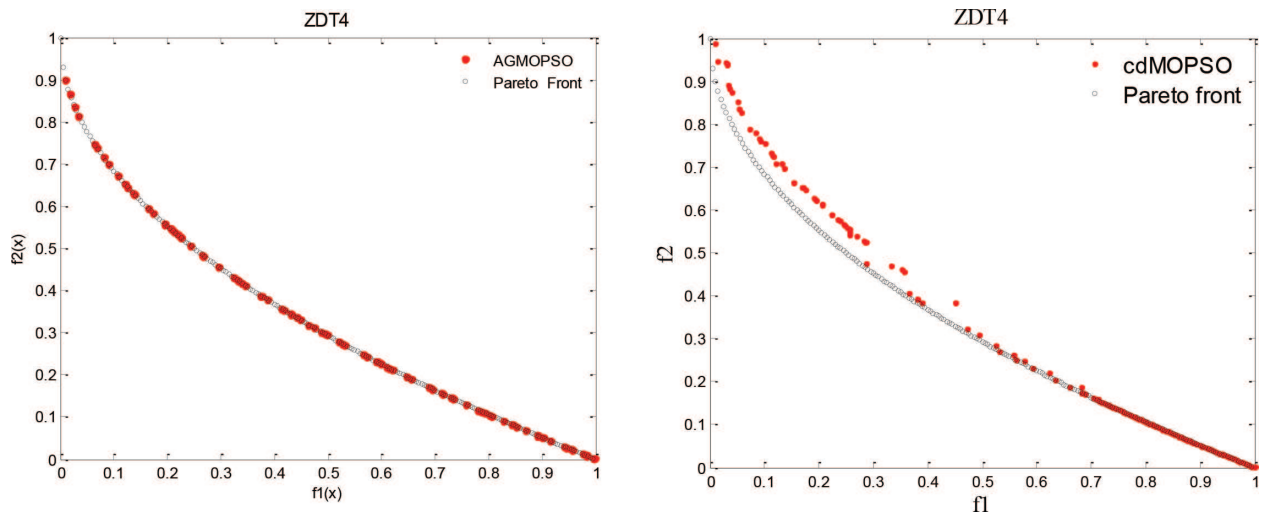


Figure 3. The Pareto front with non-dominated solutions obtained by the two multiobjective algorithms for ZDT4 function.

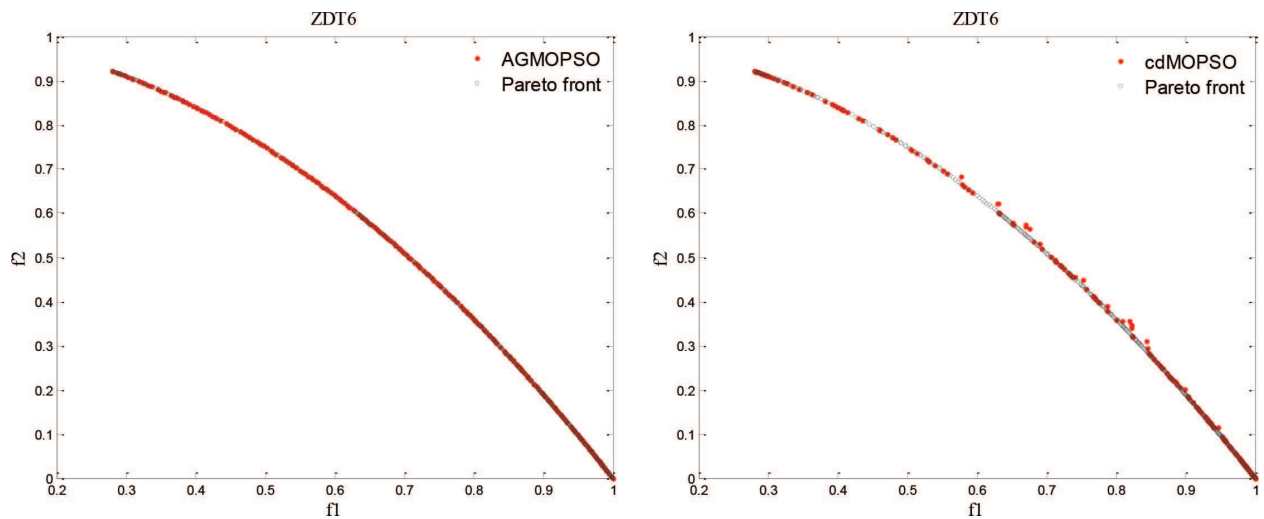


Figure 4. The Pareto front with non-dominated solutions obtained by the two multiobjective algorithms for ZDT6 function.

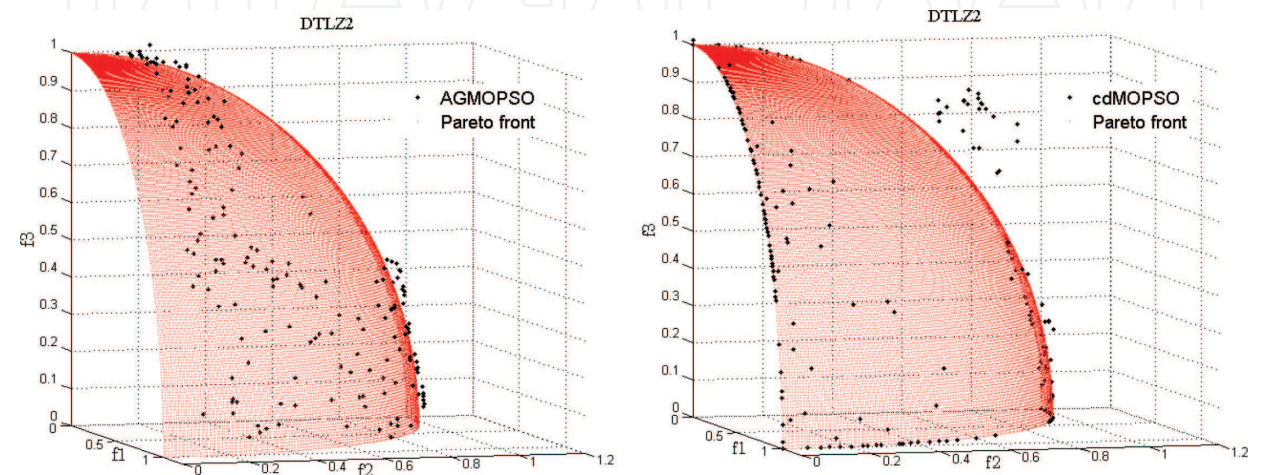


Figure 5. The Pareto front with non-dominated solutions obtained by the two multiobjective algorithms for DTLZ2 function.

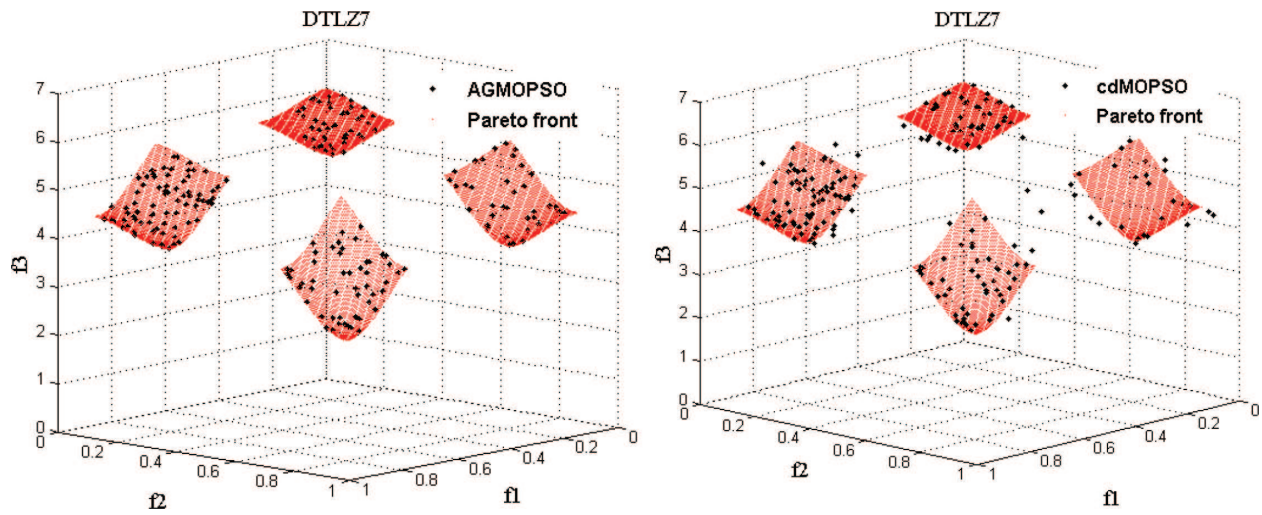


Figure 6. The Pareto front with non-dominated solutions obtained by the two multiobjective algorithms for DTLZ7 function.

Function	Index	AGMOPSO	AMOPSO	pdMOPSO	cdMOPSO	NSGA-II
ZDT3	Best	0.00149	0.00425	0.2019	0.003109	0.005447
	Worst	0.00697	0.00832	0.4265	0.028986	0.006105
	Mean	0.00433	0.00632	0.3052	0.003063	0.005834
	std	0.00297	0.00527	0.1003	0.007131	0.000202
ZDT4	Best	3.0194	2.7133	3.3980	4.9760	0.00462
	Worst	5.1522	5.0543	4.9760	6.3610	0.11166
	Mean	3.7933	3.8943	4.0330	5.9120	0.016547
	std	1.5133	2.7401	1.6510	4.5180	0.031741
ZDT6	Best	0.2046	0.0936	2.2310	0.000897	0.01119
	Worst	0.7834	0.9154	2.8790	0.003627	0.01498
	Mean	0.4878	0.5433	2.4690	0.002988	0.01286
	std	0.0242	0.0236	0.8169	0.0001543	0.001004
DTLZ2	Best	0.0477	0.0519	0.1330	0.0322	0.07830
	Worst	0.3913	0.3425	0.3690	0.2067	0.2740
	Mean	0.1058	0.1878	0.2070	0.1015	0.1059
	std	0.0060	0.0132	0.0413	0.0134	0.008383
DTLZ7	Best	0.05766	0.02044	0.00796	0.00701	0.00614
	Worst	0.32803	0.10295	0.07678	0.05439	0.03208
	Mean	0.01985	0.04573	0.04831	0.02856	0.01799
	std	0.00139	0.00312	0.00289	0.00165	0.00129

Table 2. Comparisons of different algorithms for *IGD*.

Function	Index	AGMOPSO	AMOPSO	pdMOPSO	cdMOPSO	NSGA-II
ZDT3	Best	0.023475	0.097811	0.099654	0.10356	0.081569
	Worst	0.087874	0.416626	0.487126	0.87449	0.106568
	Mean	0.067451	0.245931	0.198551	0.59684	0.092216
	std	0.012873	0.050937	0.079442	0.22468	0.008415
ZDT4	Best	0.030914	0.039825	0.069564	0.139577	0.031393
	Worst	0.078011	0.193765	0.233794	0.300951	0.044254
	Mean	0.049923	0.078821	0.186698	0.204573	0.038378
	std	0.001092	0.004517	0.063757	0.095562	0.003837
ZDT6	Best	0.010981	0.008739	0.009935	0.012396	0.006851
	Worst	0.100551	0.088535	0.023766	0.040205	0.010127
	Mean	0.034127	0.040251	0.010683	0.034569	0.008266
	std	0.009756	0.007341	0.003021	0.003884	0.000918
DTLZ2	Best	0.1438	0.0943	0.0569	0.0932	0.021456
	Worst	0.6893	0.8947	0.6991	0.5897	0.7314
	Mean	0.0398	0.4631	0.4721	0.3562	0.4162
	std	0.00764	0.03401	0.02964	0.01772	0.03655
DTLZ7	Best	0.1958	0.1047	0.0932	0.1347	0.0632
	Worst	0.9032	0.9355	0.8361	0.9307	0.7466
	Mean	0.0502	0.0493	0.4459	0.5972	0.4191
	std	0.01097	0.03201	0.00896	0.2133	0.00796

Table 3. Comparisons of different algorithms for *SP*.

a greater diversity than other compared algorithms. Experimental results in **Figures 2–4** show that the proposed AGMOPSO algorithm is superior to the cdMOPSO algorithm in diversity performance and can approach the Pareto Front. In addition, the results in **Figures 5 and 6** show that the proposed AGMOPSO algorithm can obtain a better performance on the three-objective benchmark problems with accurate convergence and the preferable diversity.

In order to show the experimental performance in details, the experimental results, which contain the best, worst, mean and standard deviations of *IGD* and *SP* based on the two-objective of ZDTs and the three-objective of DTLZs are listed in **Tables 2 and 3**, respectively.

Moreover, the experimental results in **Tables 2 and 3** include the details of the four evolutionary algorithms. To illustrate the significance of the findings, the comparing results for the performance index is analyzed as follows:

1. Comparison of *IGD* index: From **Table 2**, the proposed AGMOPSO algorithm is superior to other MOPSO algorithms in terms of the results of *IGD*. Firstly, in the two-objective of ZDTs instances, the AGMOPSO can have better mean deviations of *IGD* than other four evolutionary algorithms on ZDT3 and ZDT4. It is indicated that the MOG method has

played a vital role on the algorithm. Meanwhile, compared with NSGA-II [11], the proposed AGMOPSO has better *IGD* index performance of accuracy and stability for the two-objective of ZDTs (except ZDT4). Second, in the three-objective of DTLZs instances, the AGMOPSO is superior to other four algorithms in terms of the mean deviations value of *IGD*. According to the comparisons between the AGMOPSO and other four evolutionary algorithms, it is demonstrated that the proposed AGMOPSO is the closest to the true front and nearly enclose the entire front, which means the proposed AGMOPSO algorithm achieves the best convergence and divergence.

2. Comparison of *SP* index: The comparison of *SP* among the proposed AGMOPSO algorithm and other compared algorithms was shown in **Table 3**. Firstly, in the two-objective of ZDTs instances, the AGMOPSO can have better mean deviations and best deviations of *SP* than other four evolutionary algorithms ZDT3 and ZDT4. Meanwhile, compared with NSGA-II [11], the proposed AGMOPSO has better *SP* index performance of diversity for the two-objective of ZDTs (except ZDT6). From the results in **Table 3**, the comparison of the *SP* between the proposed AGMOPSO algorithm illustrate that the MOG method can have better effect on the diversity performance than other existing methods. Secondly, in the three-objective of DTLZs instances, the proposed AGMOPSO algorithm has the best *SP* performance on the DTLZ2 and DTLZ7 than the other four compared algorithms. In addition, to verify the effect of the MOG method, the proposed AGMOPSO can obtain a set of non-dominated solutions with greater diversity and convergence than NSGA-II on instances (except ZDT4 and ZDT6). Therefore, the proposed AGMOPSO algorithm can obtain more accurate solutions with better diversity on the most ZDTs and DTLZs.

5. Conclusion

A novel method, named AGMOPSO, is proposed to solve MOPs, which underlies MOG to accelerate the solution convergence and deletes the redundant solutions in the archive by the equidistant partition principle. Meanwhile, the convergence analysis and convergence conditions of AGMOPSO are also carefully investigated for the successful applications. Based on the theoretical analysis and the experimental results, the proposed AGMOPSO algorithm with the local search strategy MOG is a novel method for solving these MOPs. The comparisons of the different indexes also demonstrate that the proposed AGMOPSO algorithm is superior to the other algorithms for most of ZDTs and DTLZs.

Author details

Hong-Gui Han^{1,2*}, Lu Zhang^{1,2} and Jun-Fei Qiao^{1,2}

*Address all correspondence to: recharhdhan@sina.com

1 College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China

2 Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing, China

References

- [1] Wang Y, Li HX, Yen GG, Song W. MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. *IEEE Transactions on Cybernetic's*. 2015;**45**(4):830-843
- [2] Yuan W, You XG, Xu J, et al. Multiobjective optimization of linear cooperative spectrum sensing: Pareto solutions and refinement. *IEEE Transactions on Cybernetics*. 2016;**46**(1): 96-108
- [3] Brockhoff D, Zitzler E. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation*. 2009;**17**(2):135-166
- [4] Hu XB, Wang M, Paolo ED. Calculating complete and exact Pareto front for multiobjective optimization: A new deterministic approach for discrete problems. *IEEE Transactions on Cybernetics*. 2013;**43**(3):1088-1101
- [5] Martin D, Rosete A, Alcalá FJ, Francisco H. A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules. *IEEE Transactions on Evolutionary Computation*. 2014;**18**(1):54-69
- [6] Jiang SY, Yang SX. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. *IEEE Transactions on Cybernetics*. 2016;**46**(2):421-437
- [7] Nag K, Pal NR. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. *IEEE Transactions on Cybernetics*. 2016;**46**(2): 499-510
- [8] He D, Wang L, Yu L. Multi-objective nonlinear predictive control of process systems: A dual-mode tracking control approach. *Journal of Process Control*. 2015;**25**:142-151
- [9] Zhang X, Tian Y, Jin Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*. 2015;**19**(6):761-776
- [10] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*. 1994;**2**(3):221-248
- [11] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 2002;**6**(2):182-197
- [12] Zitzler E, Thiele L. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. *Tech. Rep. TIK-Report*. Zurich, Switzerland: Swiss Federal Institute Technology (ETH); 1998
- [13] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. *Tech. Rep. TIK-Report 103*. Zurich, Switzerland: Swiss Federal Institute Technology (ETH); 2001
- [14] Knowles JD, Corne DW. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*. 2000;**8**(2):149-172

- [15] Corne DW, Knowles JD, Oates MJ. The Pareto envelope-based selection algorithm for multiobjective optimization. *International Conference on Parallel Problem Solving from Nature*. 2000:839-848
- [16] Qu BY, Suganthan P, Das S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*. 2013;**17**(3): 387-402
- [17] Ali H, Shahzad W, Khan FA. Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *Applied Soft Computing*. 2012;**12**(7):1913-1928
- [18] AlRashidi MR, El-Hawary ME. A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation*. 2009;**13**(4):913-918
- [19] Elhossini A, Areibi S, Dony R. Strength pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evolutionary Computation*. 2010;**18**(1):127-156
- [20] Zhang Y, Gong D, Zhang J. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*. 2013;**103**:172-185
- [21] Li J, Zhang JQ, Jiang CJ, Zhou MC. Composite particle swarm optimizer with historical memory for function optimization. *IEEE Transactions on Cybernetics*. 2015;**45**(10):2350-2363
- [22] Hu WW, Tan Y. Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification. *IEEE Transactions on Cybernetics*. 2016;**46**(12):2719-2731
- [23] Xue B, Zhang MJ, Browne WN. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*. 2013;**43**(6): 1656-1671
- [24] Hu M, Weir JD, Wu T. An augmented multi-objective particle swarm optimizer for building cluster operation decisions. *Applied Soft Computing*. 2014;**25**:347-359
- [25] Zheng YJ, Ling HF, Xue JY, Chen SY. Population classification in fire evacuation: A multiobjective particle swarm optimization approach. *IEEE Transactions on Evolutionary Computation*. Feb. 2014;**18**(1):70-81
- [26] Lee KB, Kim JH. Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots. *IEEE Transactions on Evolutionary Computation*. 2013;**17**(6):755-766
- [27] Al Moubayed N, Petrovski A, McCall J. D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary Computation*. 2014;**22**(1):47-77
- [28] Zhu Q, Lin Q, Chen W, et al. An external archive-guided multiobjective particle swarm optimization algorithm. *IEEE Transactions on Cybernetics*. To be published. DOI: 10.1109/TCYB. 2017.2710133

- [29] Zheng YJ, Chen SY. Cooperative particle swarm optimization for multiobjective transportation planning. *Applied Intelligence*. 2013;**39**(1):202-216
- [30] Agrawal S, Panigrahi BK, Tiwari MK. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. *IEEE Transactions on Evolutionary Computation*. 2008;**12**(5):529-541
- [31] Yen GG, Leong WF. Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. 2009;**39**(4):890-911
- [32] Helwig S, Branke J, Mostaghim S. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*. 2013;**17**(2):259-271
- [33] Zhang Y, Gong DW, Ding Z. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. *Information Sciences*. 2012;**192**(1):213-227
- [34] Hu W, Yen GG. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on Evolutionary Computation*. 2015;**19**(1):1-18
- [35] Li X, Yao X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation*. 2012;**16**(2):210-224
- [36] Daneshyari M, Yen GG. Cultural-based multiobjective particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2011;**41**(2):553-567
- [37] Britto A, Pozo A. Using reference points to update the archive of MOPSO algorithms in many-objective optimization. *Neurocomputing*. 2014;**127**(1):78-87
- [38] Chakraborty P, Das S, Roy GG. On convergence of the multi-objective particle swarm optimizers. *Information Sciences*. 2011;**181**(8):1411-1425
- [39] Garcia NJ, Olivera AC, Alba E. Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*. 2013;**17**(6):823-839
- [40] Gong M, Cai Q, Chen X. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Transactions on Evolutionary Computation*. 2014;**18**(1):82-97
- [41] Mousa AA, El-Shorbagy MA, Abd-El-Wahed WF. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. *Swarm and Evolutionary Computation*. 2012;**3**(1):1-14

