A Particle Swarm Optimization Algorithm with Variable Random Functions and Mutation

ZHOU Xiao-Jun[1, 2] YANG Chun-Hua[1] GUI Wei-Hua[1] DONG Tian-Xue[1]

Abstract The convergence analysis of the standard particle swarm optimization (PSO) has shown that the changing of random functions, personal best and group best has the potential to improve the performance of the PSO. In this paper, a novel strategy with variable random functions and polynomial mutation is introduced into the PSO, which is called particle swarm optimization algorithm with variable random functions and mutation (PSO-RM). Random functions are adjusted with the density of the population so as to manipulate the weight of cognition part and social part. Mutation is executed on both personal best particle and group best particle to explore new areas. Experiment results have demonstrated the effectiveness of the strategy.

Key words Particle swarm optimization, random functions, mutation, population density

Citation Zhou Xiao-Jun, Yang Chun-Hua, Gui Wei-Hua, Dong Tian-Xue. A particle swarm optimization algorithm with variable random functions and mutation. *Acta Automatica Sinica*, 2014, **40**(7): 1339–1347

DOI 10.3724/SP.J.1004.2014.01339

Inspired by the social behavior of bird flocking and fish schooling, the particle swarm optimization (PSO) has drawn widespread attention in the last decades[1−2]. To improve the performance of the standard PSO[3], various strategies have been presented, and these studies mainly focus on the basic control parameters, such as swarm size, inertial weight, acceleration coefficients, velocity clamping, and topological structures[4−5]. In [6−8], linearly decreasing, randomly varying, and dynamic nonlinear inertial weights were proposed, respectively, indicating that a varying inertial weight can balance the global search ability and the local search ability. In [9], a particle swarm optimizer with time-varying acceleration coefficients was introduced, and it suggested that a changing acceleration coefficient of the cognition part in a linearly decreasing way and a varying acceleration coefficient of the social part in a linearly increasing way would bring better solutions. In the meantime, acceleration coefficients varying with evolutionary states were also studied by fuzzy membership functions[10].

The PSO is sensitive to control parameter choices because not only they affect the search ability but also they influence the convergence performance of particles. As a matter of fact, the convergence analysis of standard PSO has been widely studied. In [11−13], the behavior of particles was investigated to observe the trajectories for swarms under some specified models. In general, the stability analysis of PSO is based on either dynamic system theory[14−18] or stochastic process theory[19−22], and they have provided theoretical support for the selection of parameters.

On the other hand, the effect of the random functions on PSO has not been discussed so much. We presented a PSO using the skewness of evaluation function as heuristic information to adjust the random functions[23]; however, the study was based on experience without theoretical analysis. In this paper, by studying the convergence analysis results of the standard PSO, we find that the changing of random functions, personal best and group best have the potential to improve the performance of the PSO. Based on the observation, then a particle swarm optimization algorithm with variable random functions and mutation (PSO-RM) is proposed consequently, and the experiment results testify the effectiveness of the PSO-RM.

The paper is organized as follows. Section 1 introduces the standard PSO. In the next section, the search direction of optimization algorithms and random distribution functions are analyzed; then the potential of changing random functions and varying personal best and group best to improve the performance of standard PSO is demonstrated. Section 3 presents the experiment results and discussion by comparing the PSO-RM with other PSOs. Conclusions are drawn in Section 4.

1 Standard particle swarm optimizer

As a stochastic optimization approach, similar to the genetic algorithm, particles swarm optimization is initialized with a population of random solutions. However, the solutions are called particles instead of individuals, and at the same time, each particle is associated with a randomized velocity. In the process of updating, each particle keeps track of its previous best position as well as the best position found by its neighborhood. The standard particle swarm optimization can be described as follows

$$v_{id}(k+1) = wv_{id}(k) + c_1r_1(p_{id}(k) - x_{id}(k)) + c_2r_2(p_{gd}(k) - x_{id}(k)) \tag{1}$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \tag{2}$$

where i denotes the ith particle, d is the dth dimension, $V_i(k) = (v_{i1}(k), \cdots, v_{in}(k))$ and $X_i(k) = (x_{i1}(k), \cdots, x_{in}(k))$ are the current velocity and position of the ith particle, respectively. $P_i(k) = (p_{i1}(k), \cdots, p_{in}(k))$ is the best position found so far by the ith particle, while $P_g(k) = (p_{g1}(k), \cdots, p_{gn}(k))$ is the best position found by the neighborhood. c_1 and c_2 are acceleration constants, and r_1 and r_2 are uniformly distributed random numbers.

2 PSO with variable random functions and mutation

2.1 Search direction

To solve an optimization problem, the traditional method is based on gradient. In the gradient based way,

Manuscript received May 30, 2011; accepted November 21, 2013
Supported by National Natural Science Found for Distinguished Young Scholars of China (61025015), the Foundation for Innovative Research Groups of National Natural Science Foundation of China (61321003) and the China Scholarship Council
Recommended by Associate Editor WANG Hong-Wei
1. School of Information Science and Engineering, Central South University, Changsha 410083, China 2. School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia

search direction is generated by the negative gradient at an iterative point. Moving in the negative gradient direction, the objective function decreases at a steepest descent in the neighborhood of the point. However, the gradient only indicates local information, that is to say, methods based on gradient are local search approach to some extent.

Modern intelligent optimization algorithms, such as genetic algorithm, simulated annealing, ant colony optimization, particle swarm optimization, etc., are believed to be global optimization methods. At the same time, these algorithms are stochastic approaches, and they usually utilize the evaluation function as the search direction information. In a way, for the intelligent optimization algorithm, the probability distribution of population can reflect the direction to guide the swarm′s moving. As for the PSO algorithm, the changing of the random functions in the update equation can alter the swarm′s distribution, which, in other words, can direct the behavior of particles.

2.2 Random distribution functions

In various distribution functions, the β distribution can approximate other diverse distribution functions. The probability density function of β distribution is

$$f(x|a,b)=\frac{1}{B(a,b)}x^{a-1}(1-x)^{b-1},\ 0<x<1 \tag{3}$$

$$B(a,b)=\int_0^1 x^{a-1}(1-x)^{b-1}\mathrm{d}x,\ a>0,\ b>0 \tag{4}$$

The alteration of parameters a and b can gain various distributions. If $a=b=1$, the distribution can approximate uniform distribution; if $a=b=4$, it will approximate normal distribution; if $a=1, b=2$, the approximate lower triangular distribution will be gained and if $a=2, b=1$, the approximate upper triangular distribution will be obtained. The histograms of the distributions are illustrated in Fig. 1.

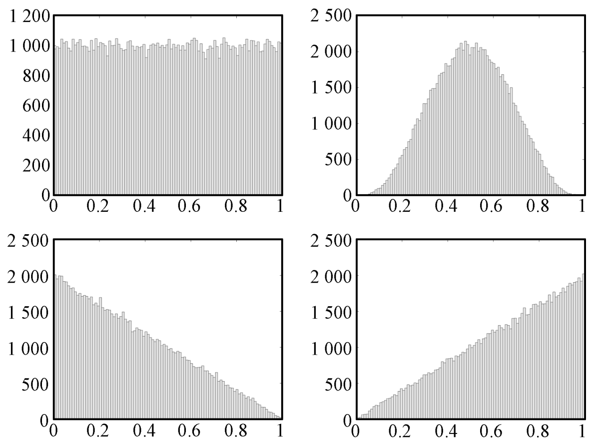


Fig. 1 Four different distributions

2.3 Adjusting strategy of random functions

In [18], the convergence analysis of PSO was discussed. It can be concluded that under some circumstances, the positions of the particles can converge to

$$q_{id}=\frac{\varphi_1 p_{id}+\varphi_2 p_{gd}}{\varphi_1+\varphi_2} \tag{5}$$

where $\varphi_1=c_1r_1, \varphi_2=c_2r_2$. It is acknowledged that p_{id} is the cognition part and p_{gd} is the social part, and the equilibrium of the two parts is manipulated by φ_1 and φ_2. It is easy to understand that the increase of φ_1 will benefit competition while the increase of φ_2 will benefit cooperation. To control the competition and cooperation, it is very significant to evaluate the situation of the population distribution information.

The diversity of the population is an efficient way to evaluate the distribution. In [24], various methods based on distance were presented to measure the population diversity. In this paper, the relative density of population is used for the same purpose.

$$\rho_{abs}=\frac{m}{V}=\frac{m}{L_1\times L_2\times\cdots\times L_D} \tag{6}$$

where ρ_{abs} is the absolute density of the population, and $L_d(d=1,\cdots,D)$ is the range of the dth dimensional particles, and m is the size of the population. Generally, the size of population is constant, so it is useful to define the relative density

$$\rho_{rel}=\frac{L_1\times L_2\times\cdots\times L_D}{L'_1\times L'_2\times\cdots\times L'_D}\in[0,1] \tag{7}$$

in which the ranges of L'_1 to L'_D are specified as the difference between the upper and lower bounds of variables. Due to the ρ_{rel}′s variation with the dimensions of the variables, the single dimensional analysis is beneficial for deep study.

$$\rho_{rel}(d)=\frac{L_d}{L'_d}\in[0,1] \tag{8}$$

In the paper, acceleration constants are fixed; therefore, the changing of random functions will control the weight of competition and cooperation, and then affect the distribution of population. Using the information of density for criterion, the detailed relationship is specified as follows

$$\begin{cases}r_1=f(x|2,1), r_2=f(x|1,2);\ \text{if}\ \rho_{rel}(d)>0.75\\ r_1=f(x|4,4), r_2=f(x|4,4);\ \text{if}\ 0.5<\rho_{rel}(d)\le 0.75\\ r_1=f(x|1,1), r_2=f(x|1,1);\ \text{if}\ 0.25<\rho_{rel}(d)\le 0.5\\ r_1=f(x|1,2), r_2=f(x|2,1);\ \text{if}\ \rho_{rel}(d)\le 0.25\end{cases} \tag{9}$$

The reason why we choose 0.25, 0.5 and 0.75 as thresholds is that the expected values of both the uniform distribution and standard normal distribution which are widely used for PSO are 0.5, while 0.25 and 0.75 are the mean values in $[0, 0.5]$ and $[0.5, 1]$, respectively.

It indicates that when the relative density is high, the particles tend to their own best, and at last, the particles tend to the group best with a rapid convergence rate.

2.4 Mutation of personal best and group best

To maintain the diversity of a population, mutation operators are widely used in evolutionary computation, especially for PSO. Different mutation operators for PSO are investigated and compared in [25], including Gaussian operator, Cauchy operator, Michalewicz operator, and Random operator. At the same time, three different ways to carry out the operators during an optimization run were also summarized. And it was concluded that mutation on best position found by the neighborhood was a common choice and the best mutation method depended on the structure of optimization problem.

In the specified parameters of β distribution, the expectations of random numbers are

$$\begin{cases}\mathrm{E}(f(x|1,1))=0.5, \mathrm{E}(f(x|1,2))=\frac{1}{3}\\ \mathrm{E}(f(x|2,1))=\frac{2}{3}, \mathrm{E}(f(x|4,4))=0.5\end{cases} \tag{10}$$

which shows that the expectation of $r_1 + r_2$ will be lower than 2 in any case, which can guarantee the convergence property of the proposed PSO.

On the other hand, as shown in (5), the personal best and group best are not fixed actually in the process of searching. However, the phenomenon is ignored by most of the researchers as for the simplicity of the convergence analysis. Out of question, the emphasis on convergence will weaken the global search.

To expand the search space and enhance the search capability, the changing of the two bests becomes necessary. Although a learning method using other particles′ historical information was used to update the best[26], the strategy has no theoretical basis. Of course, unlike other existing methods for PSO, the polynomial mutation[27] is performed on them in the paper as follows

$$y_k = x_k + (x_k^u - x_k^l)\delta_k \tag{11}$$

where y_k is the child and x_k is the parent with x_k^u being the upper bound on the parent component, x_k^l is the lower bound and δ_k is a random number computed from a polynomial distribution by

$$\begin{cases} \delta_k = (2r_k)^{\frac{1}{\eta_m+1}}, \text{if} \quad r_k < 0.5 \\ \delta_k = 1 - [2(1-r_k)]^{\frac{1}{\eta_m+1}}, \text{if} \quad r_k \geq 0.5 \end{cases} \tag{12}$$

in which r_k is an uniformly random number between (0, 1) and η_m is the mutation distribution index.

2.5 Description of the PSO-RM

The differences from the standard version of PSO are the variable random functions and mutation. In this paper, the strategy of deceasing inertial weight is inherited, and random functions will change according to the relative density of the population. By the way, the personal best and group best after mutation will be incorporated with original bests through selection by sorting the fitness and only half of the top will be chosen.

The PSO-RM in pseudocodes is given as follows:

Initialize population, set $w \leftarrow w_{\max}$ and $k \leftarrow 0$;
repeat
$k \leftarrow k + 1$
Adjust random functions in terms of (9);
Mutate *pbest* and *gbest* through (11) and (12);
Select *pbest* and *gbest* by sorting
$w \leftarrow w_{\max} - (w_{\max} - w_{\min})\frac{k}{\text{MAXITER}}$;
until the specified termination criterion is met.

where *pbest* and *gbest* are short for personal best and group best, $w_{\max}$ and $w_{\min}$ are the maximal and minimal weights, and MAXITER is a predefined maximum number of iterations. To be more specific, inertial weight w varies from 0.9 to 0.4, and the mutation distribution index η_m in (12) is 20, while the probability of mutation is the reciprocal of the dimension depending on specified problems.

3 Experiments and discussion

To evaluate the performance of the PSO-RM, a set of twenty well-known benchmark functions are used, including both unimodal and multimodal functions, non-rotated or non-shifted and rotated or shifted. The majority of non-rotated and non-shifted functions are chosen from [28], the rotated and shifted functions are from the first 10 benchmark functions in [29], and the details of these functions are displayed in Table 1.

Table 1 Benchmark functions used in this paper

Function	Range
$f_1(\boldsymbol{x}) = \sum_{i=1}^n x_i^2$	[−100, 100]
$f_2(\boldsymbol{x}) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[−30, 30]
$f_3(\boldsymbol{x}) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	[−5.12, 5.12]
$f_4(\boldsymbol{x}) = \frac{1}{4\,000}\sum_{i=1}^n x_i^2 - \prod_i^n \cos\lvert\frac{x_i}{\sqrt{i}}\rvert + 1$	[−600, 600]
$f_5(\boldsymbol{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	[−32, 32]
$f_6(\boldsymbol{x}) = \sum_{i=1}^n \sin(x_i)\sin(\frac{ix_i^2}{\pi})^{20}$	[−π, π]
$f_7(\boldsymbol{x}) = \sum_{i=1}^n (-x_i\sin(\sqrt{\lvert x_i\rvert}))$	[−500, 500]
$f_8(\boldsymbol{x}) = \sum_{i=1}^n \lvert x_i\rvert + \prod_{i=1}^n \lvert x_i\rvert$	[−10, 10]
$f_9(\boldsymbol{x}) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	[−100, 100]
$f_{10}(\boldsymbol{x}) = \max_i\{\lvert x_i\rvert, 1 \leq i \leq n\}$	[−100, 100]
$f_{11}(\boldsymbol{x}) = \sum_{i=1}^n z_i^2, \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	[−100, 100]
$f_{12}(\boldsymbol{x}) = \sum_{i=1}^n (\sum_{j=1}^i z_j)^2, \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	[−100, 100]
$f_{13}(\boldsymbol{x}) = \sum_{i=1}^n (10^6)^{\frac{i-1}{n-1}} z_i^2, \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M}$	[−100, 100]
$f_{14}(\boldsymbol{x}) = (\sum_{i=1}^n (\sum_{j=1}^i z_j)^2) * (1 + 0.4\lvert N(0,1)\rvert), \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M}$	[−100, 100]
$f_{15}(\boldsymbol{x}) = \max\{\lvert \boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{B}_i\rvert\}, \boldsymbol{B}_i = \boldsymbol{A}_i * \boldsymbol{o}$	[−100, 100]
$f_{16}(\boldsymbol{x}) = \sum_{i=1}^n (100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2), \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o} + \boldsymbol{1}$	[−100, 100]
$f_{17}(\boldsymbol{x}) = \frac{1}{4\,000}\sum_{i=1}^n z_i^2 - \prod_i^n \cos\lvert\frac{z_i}{\sqrt{i}}\rvert + 1, \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M}$	[0, 600]
$f_{18}(\boldsymbol{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n z_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi z_i)) + 20 + e, \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M}$	[−32, 32]
$f_{19}(\boldsymbol{x}) = \sum_{i=1}^n (z_i^2 - 10\cos(2\pi z_i) + 10), \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	[−5, 5]
$f_{20}(\boldsymbol{x}) = \sum_{i=1}^n (z_i^2 - 10\cos(2\pi z_i) + 10), \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M}$	[−5, 5]

The algorithms which are compared with the proposed PSO-RM are listed below:

1) the standard PSO (PSO);
2) PSO with only variable random functions (PSO-R);
3) PSO with only mutation on *pbest* and *gbest* (PSO-M);
4) Comprehensive learning PSO (CLPSO);
5) PSO-RM.

In PSO, the inertial weight varies in a linear declining way from 0.9 to 0.4 and both acceleration coefficients are fixed at 2. In PSO-R and PSO-M, the parameter settings are the same as PSO except the random functions and the mutation strategy, while in CLPSO, the parameter setting in accordance with [26] is used. In the proposed PSO-RM, parameters are described in the last part.

The population size has some effect on the performance of the PSOs; however, it is quite common to set the number of particles to the range from 20 to 60[4]. In this paper, all experiments are carried out with a population size of 30, and they are all performed independently for 30 trials with the dimension 30 and the maximum iterations at 1×10^5. By the way, all of the simulations are run on the MATLAB platform with version 7.1.0.246 (R14) Service Pack 3. The test results of the benchmark functions are presented in Table 2, with mean and s.t. (standard deviation) to indicate their performance. At the same time, t-distribution tests are used to describe the significant differences of results obtained by PSO-RM and the other variants of PSO, as shown in Table 2, in which, the significance level at 5 % is set in the paper. "$-$", "+", and "$\approx$" mean that the performances of the corresponding algorithms are worse, better and similar to that of PSO-RM, respectively.

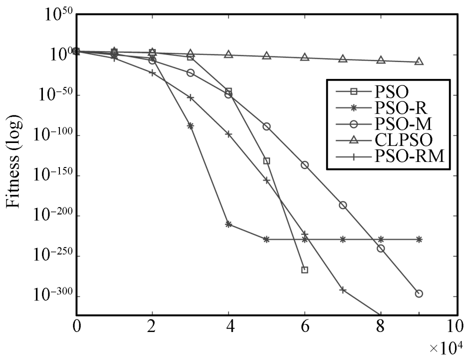


Fig. 2 Mean fitness for f_1

1) Non-rotated and non-shifted functions f_1-f_{10}: It is easy to find that the PSO-RM is the best among the five methods on these non-rotated and non-shifted functions except for f_4, for which the CLPSO dominates the others. Compared PSO-R with PSO, it is not difficult to find that PSO-R is better than PSO for f_2, f_8, f_9, f_{10}, while for others, they behave much the same. As for PSO-M, we observe that it beats PSO on all the cases and it has the similar behavior to that of PSO-RM, that is to say, it indicates that the success of PSO-RM is mainly due to the mutation on *pbest* and *gbest*. On the other hand, when comparing PSO-M with PSO-R, as shown from Fig. 3 to Fig. 12, we find that the curves of PSO-R decrease more sharply than that of PSO-M, which indicates that the strategy of changing the variable random functions accelerate the convergence rate of the standard PSO. By the way, for the Rosenbrock function (f_2), as far as we know, the PSO-RM achieves the best results compared with most variants of PSO in other literatures. Any way, the good results are due to the combination of the mutation and the variable random functions strategy.

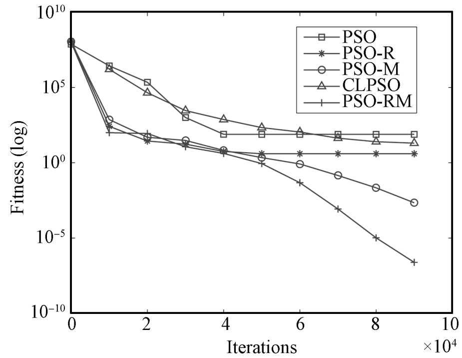


Fig. 3 Mean fitness for f_2

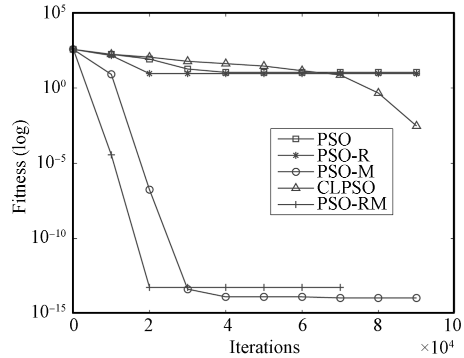


Fig. 4 Mean fitness for f_3

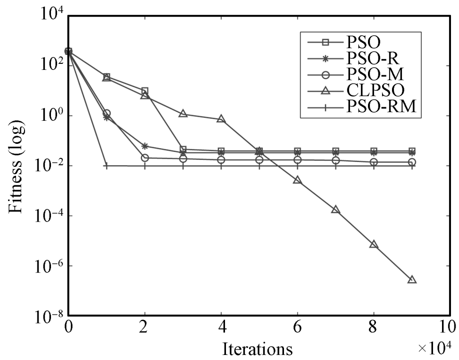


Fig. 5 Mean fitness for f_4

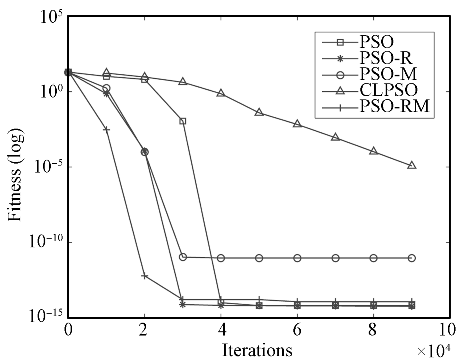


Fig. 6 Mean fitness for f_5

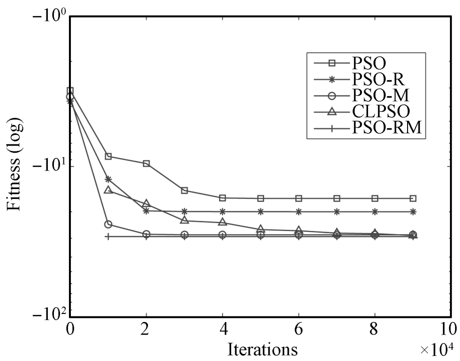


Fig. 7 Mean fitness for f_6

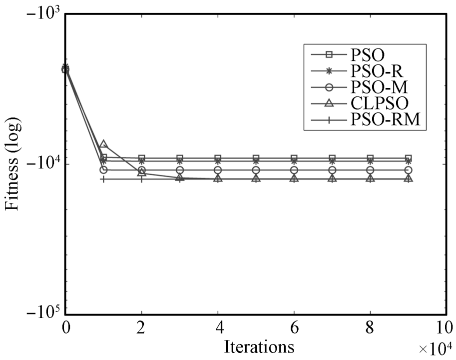


Fig. 8 Mean fitness for f_7

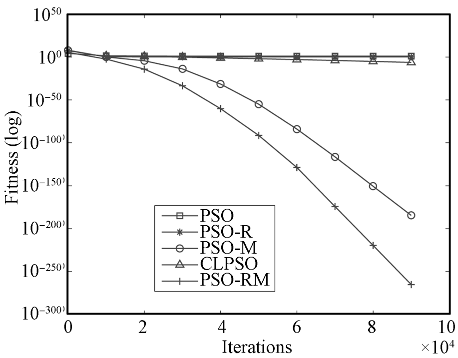


Fig. 9 Mean fitness for f_8

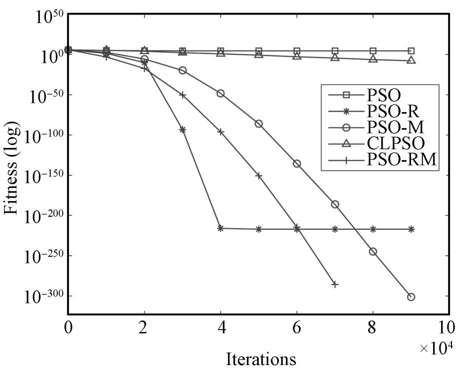


Fig. 10 Mean fitness for f_9

2) Rotated and shifted functions f_{11}-f_{20}: As can be seen from Table 2, apart from f_{11}, f_{12} and f_{19}, all of the results of the PSOs are not satisfactory, especially for the standard PSO. However, compared with its competitors, the results of PSO-RM are much better. To be more specific, for f_{12} and f_{14}, the results of PSO-RM are more than a little satisfactory when compared with that of CLPSO. Furthermore, we can find the same phenomenon as described previously, that is, the polynomial mutation of *pbest* and *gbest* contributes much to the performance of PSO-RM, and the strategy of variable random functions can accelerate the convergence process. We can also find that the proposed PSO-RM is good for shifted functions, but when running with rotated functions, it becomes not so satisfactory. For f_{13} and f_{15}, the results are bad for all variants of PSO.

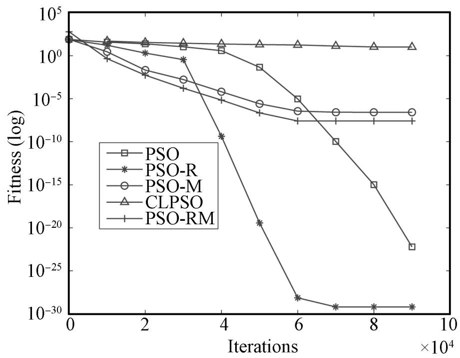


Fig. 11 Mean fitness for f_{10}

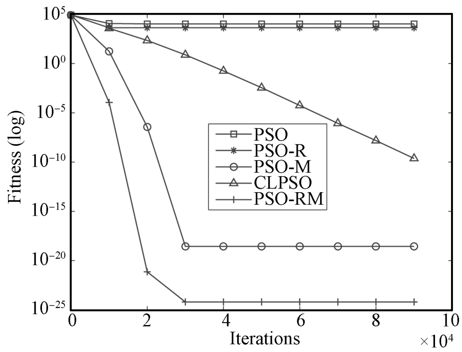


Fig. 12 Mean fitness for f_{11}

Then, we explain the reasons of the success of PSO-RM for some cases.

As a stochastic algorithm, whether or not to find a global minimum depends mainly on how to update a population. In theory, if a population can keep changing all the way, it has the ability to find the global minimum but in a infinite time. However, to make a stochastic algorithm efficient, strategies have to be proposed to reduce the large amount of time. As the number of iterations is fixed, the rapid convergence rate is necessary in practice. In the PSO-RM, the variable random function strategy has the ability to better explore at the first stage and to better exploit at the last stage, which consequently leads to the rapid convergence in a finite time. Nevertheless, if the *pbest* and *gbest* are local minimum not global minimum and are constant for a long time, they will become blocked in local minima, from which the population are difficult to escape. Then other strategy of mutation on *pbest* and *gbest* is adopted, so as to benefit the exploration of new areas, which improves the performance of the standard PSO.

Table 2 Test results of the benchmark functions

Fcn	statistics	PSO	PSO-R	PSO-M	CLPSO	PSO-RM
f_1	mean	0	9.38E−230	4.94E−324	1.69E−11	0
	s.t.	0	0	0	8.48E−12	0
	t test	≈	−	≈	−	
f_2	mean	19.0714	3.9866	1.68E−4	22.0831	6.62E−10
	s.t.	25.157	8.0141	7.03E−4	17.2449	1.86E−9
	t test	−	−	−	−	
f_3	mean	12.6028	8.9546	1.13E−14	7.96E−5	3.22E−14
	s.t.	3.9827	13.0235	2.75E−14	9.47E−5	8.66E−14
	t test	−	−	≈	−	
f_4	mean	0.0231	0.0330	0.0139	5.56E−8	0.0094
	s.t.	0.0313	0.0338	0.0128	7.23E−8	0.0096
	t test	−	−	−	+	
f_5	mean	6.09E−15	5.63E−15	8.97E−12	1.47E−6	1.13E−14
	s.t.	6.48E−16	1.35E−15	4.90E−11	3.45E−7	9.12E−15
	t test	≈	≈	−	−	
f_6	mean	−16.5759	−19.9847	−28.4591	−29.0535	−29.2183
	s.t.	1.9788	1.7189	1.1324	0.1491	0.4962
	t test	−	−	≈	≈	
f_7	mean	−9.02E+3	−9.54E+3	−1.09E+4	−1.25E+4	−1.25E+4
	s.t.	783.7548	523.3748	372.4628	30.0488	48.1852
	t test	−	−	−	≈	
f_8	mean	10	1.6667	1.98E−219	7.16E−8	0
	s.t.	9.0972	4.6113	0	2.29E−8	0
	t test	−	−	−	−	
f_9	mean	2.63E+4	1.09E−217	3.46E−323	2.15E−10	0
	s.t.	2.87E+4	0	0	1.51E−10	0
	t test	−	−	≈	−	
f_{10}	mean	2.08E−27	6.27E−30	2.67E−7	8.9718	2.57E−8
	s.t.	5.39E−27	2.53E−29	1.09E−6	1.1409	6.29E−8
	t test	+	+	≈	−	
f_{11}	mean	1.02E+4	4.14E+3	2.80E−19	3.71E−12	6.39E−25
	s.t.	3.90E+3	2.21E+3	1.53E−18	1.90E−12	3.18E−24
	t test	−	−	−	−	
f_{12}	mean	2.31E+4	7.11E+3	7.71E−16	4.51E+3	2.46E−23
	s.t.	1.43E+4	4.69E+3	4.22E−15	939.6842	6.28E−23
	t test	−	−	−	−	
f_{13}	mean	1.09E+8	2.88E+7	1.37E+5	2.33E+7	6.18E+4
	s.t.	1.40E+8	2.65E+7	6.01E+4	7.61E+6	2.95E+4
	t test	−	−	≈	−	
f_{14}	mean	1.92E+4	7.71E+3	0.0554	1.34E+4	0.0299
	s.t.	1.01E+4	8.33E+3	0.0889	3.41E+3	0.0330
	t test	−	−	≈	−	
f_{15}	mean	1.35E+4	9.28E+3	5.80E+3	4.98E+3	2.23E+3
	s.t.	3.98E+3	4.06E+3	1.82E+3	536.9043	972.0416
	t test	−	−	≈	≈	
f_{16}	mean	3.31E+9	2.69E+8	7.5727	31.7994	0.1326
	s.t.	4.77E+9	2.65E+7	11.8432	17.6113	0.3148
	t test	−	−	−	−	
f_{17}	mean	488.7066	284.4888	0.0180	1.1631	0.0223
	s.t.	247.2481	243.8877	0.0163	0.0696	0.0197
	t test	−	−	+	−	
f_{18}	mean	20.8079	20.8057	20.1510	20.9724	20.1365
	s.t.	0.0566	0.0606	0.1363	0.0567	0.0950
	t test	≈	≈	≈	≈	
f_{19}	mean	137.7333	91.4834	4.73E−16	1.64E−6	9.47E−16
	s.t.	26.4447	22.7671	9.25E−16	1.14E−6	1.85E−15
	t test	−	−	≈	−	
f_{20}	mean	216.4011	171.1572	242.3854	138.2069	80.9610
	s.t.	35.4296	46.1444	50.1928	15.8127	46.9782
	t test	−	−	−	−	

Next, we give some reasons why PSO-RM does not work for other cases.

For the non-rotated and non-shifted group, PSO-RM is not good for the Griewank function (f_4). From the experiments of this function, we observe that sometimes PSO-RM could find the global minimum, but at other times, it failed. In the unsuccessful cases, we find that some components of the *pbest* and *gbest* do not change until the end of the iterations. As we can see from (1) and (2), if the *pbest* and *gbest* stop changing, the random functions will have no effect on the position and velocity equations. Although the mutation is operated on the *pbest* and *gbest*, because of low mutation probability on single dimensional component, the PSO-RM gets trapped into local minimum and can hardly escape. In other words, it indicates signifies that high mutation probability is necessary for single dimensional component. In CLPSO, the best particle for guidance can be chosen randomly from one of all particles′ historical best component, which is just a case of increasing the changing of single dimensional component, that is why CLPSO is the best among all PSOs on this function. However, the strategy can only be useful for functions whose variables are independent, for example, for Rosenbrock function (f_2) and Griewank function with shift and rotation (f_{17}), it does not work. Anyhow, the success of CLPSO supports the strategy of mutation on personal best and group best.

For the rotated and shifted group, PSO-RM does work well for most cases as other variants of PSO. From Figs. 15, 16, 18 and 20, we can find that the curves of PSO-RM decrease in a slow way at the last stage of the iterative process, which can be explained by the relationship to adjust the random functions. If the relative density is low, the particles tend to the group best with a rapid rate. However, if the personal best and group best are local minima, the PSO-RM will be in stagnation. In other words, for these cases, the strategy of changing the random functions does not work, and it does not help to accelerate the convergence. From Fig. 17 and Fig. 20, the curves of PSO-RM keep unchanged at an early state, which indicates that the mutation does not work for the cases, because the polynomial mutation in the paper has little effect on single dimensional component. However, the failure of PSO-RM on f_{17} and f_{20} indicates the cooperation of random distribution functions and mutation is a potential strategy, in other words, a proper random function should be associated with a corresponding proper mutation operator.

All in all, the performance of PSO-RM is better than its competitors used in the paper, which testifies the effectiveness of the proposed strategy. It is found that the changing of random functions and mutation of personal best and group best are significant for PSO. To deal with the shifted

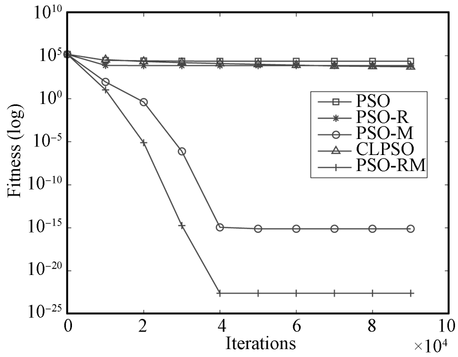


Fig. 13 Mean fitness for f_{12}

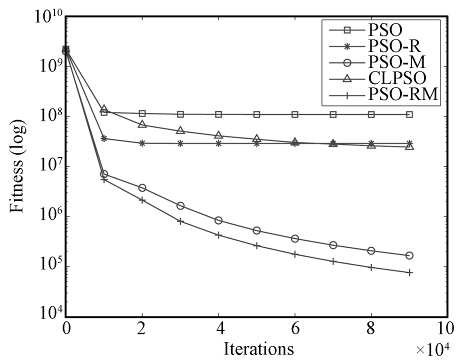


Fig. 14 Mean fitness for f_{13}

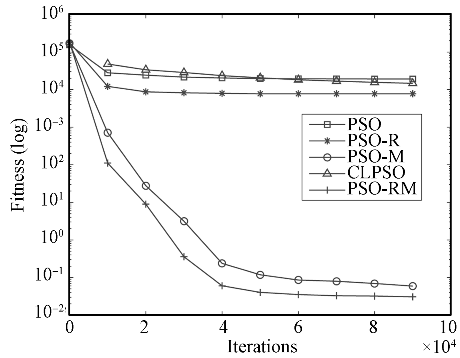


Fig. 15 Mean fitness for f_{14}

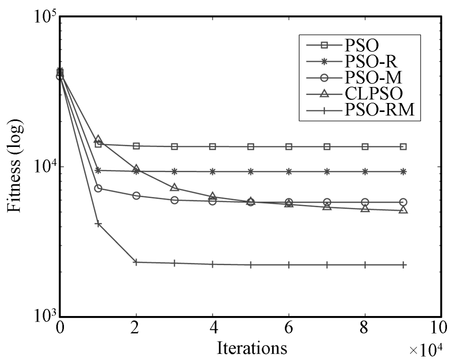


Fig. 16 Mean fitness for f_{15}

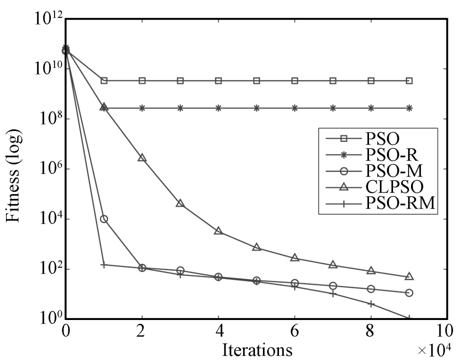


Fig. 17 Mean fitness for f_{16}

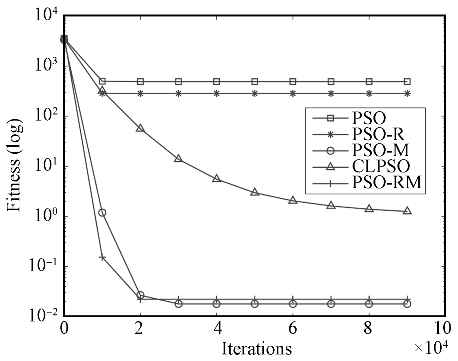


Fig. 18 Mean fitness for f_{17}

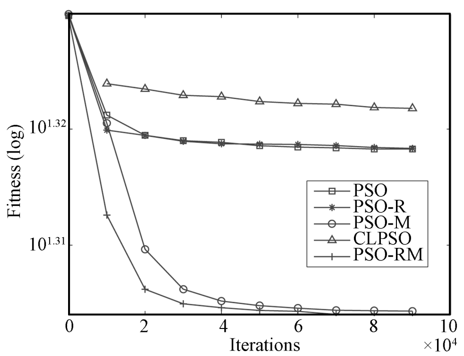


Fig. 19 Mean fitness for f_{18}

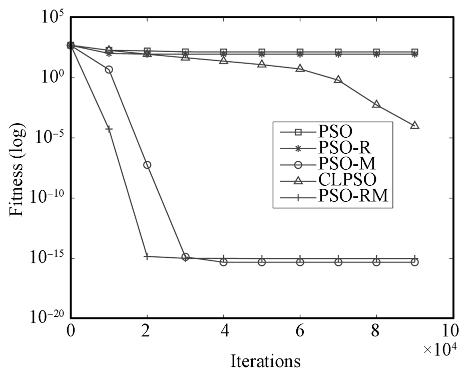


Fig. 20 Mean fitness for f_{19}

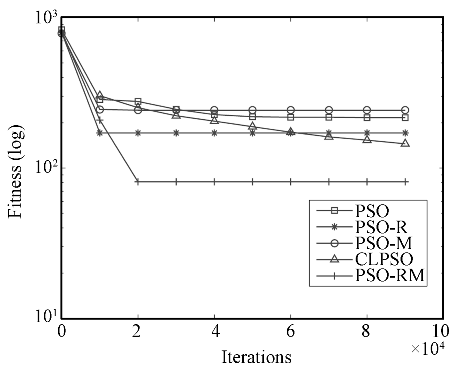


Fig. 21 Mean fitness for f_{20}

and rotated functions, not only should we focus on the features of these functions, but also we should pay attention to the iterative process of the position and velocity equations. That is to say, the convergence analysis can only guarantee the PSO converges to a stationary point but not the global minimum. To make sure the point is just the global minimum and to make PSO algorithms more efficient, other strategies need to be considered.

4 Conclusion

By studying the convergence of standard particle swarm optimization, the necessity of changing the random functions and personal and group best is found. Then, the adjusting strategy of random functions based on the relative density of a population and the polynomial mutation on personal as well as group best are proposed. Compared with other variants of PSO, the experiment results have testified the effectiveness of the proposed strategy. Especially for Rosenbrock function, the PSO-RM achieves a quite satisfactory result.

On the other hand, the PSO-RM does not work well for some shifted and rotated functions. Although it is shown that random functions, personal best and group best have great effect on PSO, how to use a better strategy to adjust the random distribution functions and to design a better mutation on best particles are still challenging. That is to say, much work has to be done for PSO, and to understand the features of these rotated functions is our future work to better improve the performance of PSO.

Acknowledgement

We really appreciate Professor P. N. Suganthan for his original MATLAB codes. We would also like to thank the reviewers for their valuable comments and suggestions that helped improve the quality of this paper.

References

1 Banks A, Vincent J, Anyakoha C. A review of particle swarm optimization Part I: background and development. *Natural Computing*, 2007, **6**(4): 467−484

2 Banks A, Vincent J, Anyakoha C. A review of particle swarm optimization Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing*, 2007, **7**(1): 109−124

3 Shi Y H, Eberhart R C. A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Anchorage, AK: IEEE, 1998. 69−73

4 Shi Y H, Eberhart R C. Paramter selection in particle swarm optimization. *Evolutionary Programming VII*. Berlin, Germany: Springer-Verlag, 1997. 591−600

5 Eberhart R C, Shi Y H. Particle swarm optimization: developments, applications and resources. In: Proceedings of IEEE International Conference on Evolutionary Computation. Seoul: IEEE, 2001, **1**: 81−86

6 Shi Y S, Eberhart R C. Empirical study of particle swarm optimization. In: Proceedings of the IEEE International Congress on Evolutionary Computation. Washington DC: IEEE, 1999, **3**: 591−600

7 Eberhart R C, Shi Y S. Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the IEEE Congress on Evolutionary Computation. Seoul, Korea: IEEE, 2001, **3**: 94−97

8 Bin J, Lian Z G, Gu X S. A dynamic inertial weight particle swarm optimization algorithm. *Chaos Solitons & Fractals*, 2008, **37**(3): 698−705

9 Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierachical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(3): 240−255

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

11 Kennedy J. The behavior of particles. In: Proceedings of the 7th International Conference on Evolutionary Programming. San Diego: IEEE, 1998. 579−589

12 van den Bergh F, Engelbrecht A P. A study of particle swarm optimization particle trajectories. *Information Sciences*, 2006, **176**(8): 937−971

13 Clerc M. Stagnation Analysis in Particle Swarm Optimisation or What Happens When Nothing Happens, Technical Report CSM-460, Department of Computer Science, University of Essex, 2006

14 Chen J, Pan F, Cai T, Tu X Y. Stability analysis of particle swarm optimization without Lipschitz constraint. *Journal of Control Theory and Applications*, 2003, **1**(1): 86−90

15 Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 2003, **85**(6): 317−325

16 Kadirkamanathan V, Selvarajah K, Fleming P J. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, 2006, **10**(3): 245−255

17 Fernández-Martínez J L, García-Gonzalo E, Fernández-Alvarez J P. Theoretical analysis of particle swarm trajectories through a mechanical analogy. *International Journal of Computational Intelligence Research*, 2008, **4**(2): 93−104

18 Clerc M, Kennedy J. The particle swarm: explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(1): 58−73

19 Jiang M, Luo Y P, Yang S Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 2007, **102**(1): 8−16

20 Chen Y P, Jiang P. Analysis of particle interaction in particle swarm optimization. *Theoretical Computer Science*, 2010, **411**(21): 2101−2115

21 Jin Xin-Lei, Ma Long-Hua, Wu Tie-Jun, Qian Ji-Xin. Convergence analysis of the particle swarm optimization based on stochastic processes. *Acta Automatica Sinica*, 2007, **33**(12): 1263−1268 (in Chinese)

22 Su Shou-Bao, Cao Xi-Bin, Kong Min. Stability analysis of particle swarm optimization using swarm activity. *Control Theory & Application*, 2010, **27**(10): 1411−1417 (in Chinese)

23 Zhou Xiao-Jun, Yang Chun-Hua, Gui Wei-Hua. Particle swarm optimization algorithm with variable random function. In: Proceedings of the 30th Chinese Control Conference. Yantai, China: IEEE, 2011. 5408−5412

24 Shi Y S, Eberhart R C. Population diversity of particle swarms. In: Proceedings of the IEEE Congress on Evolutionary Computation. Hong Kong, China: IEEE, 2008. 1063−1067

25 Andrews P S. An investigation into mutation operators for particle swarm optimization. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver, BC: IEEE, 2006. 1044−1051

26 Liang J J, Qin A K. Suganthan P N, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transaction on Evolutionary Computation*, 2006, **10**(3): 281−295

27 Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 1996, **26**(4): 30−45

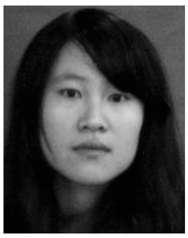
28 Yao X, Liu Y, Lin G M. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, **3**(2): 82−102

29 Suganthan P N, Hansen N, Liang J J, Deb K, Chen Y P, Auger A, Tiwari S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. Edinburgh, UK: IEEE, 2005. 1−50

ZHOU Xiao-Jun Received his bachelor′s degree in automation in 2009 from Central South University, China. He is a Ph. D. candidate at University of Ballarat, Victoria, Australia. His interest covers modeling, optimization and control of complex industrial process, optimization theory and algorithms, state transition algorithm, duality theory and their applications.
E-mail: tiezhongyu2010@gmail.com

YANG Chun-Hua Received her M. Eng. in automatic control engineering and her Ph. D. degree in control science and engineering from Central South University, China in 1988 and 2002, respectively, and was with the Electrical Engineering Department, Katholieke Universiteit Leuven, Belgium from 1999 to 2001. She is currently a full professor in the School of Information Science & Engineering, Central South University. Her research interest covers modeling and optimal control of complex industrial process, intelligent control system, and fault-tolerant computing of real-time systems. E-mail: ychh@csu.edu.cn

GUI Wei-Hua Received the degree of the B. Eng. in automatic control engineering and the M. L. in control science and engineering from Central South University, China in 1976 and 1981, respectively. From 1986 to 1988, he was a visiting scholar at Universitat-GH-Duisburg, Germany. He has been a full professor in the School of Information Science & Engineering, Central South University, China since 1991. His research interest covers modeling and optimal control of complex industrial process, distributed robust control, and fault diagnoses. E-mail: gwh@mail.csu.edu.cn



DONG Tian-Xue Received her bachelor degree in automation from Central South University, China in 2013. She is a master student in control science and engineering from Central South University, China, since 2013. Her research interest covers discrete optimization, mixed integer linear programming, intelligent optimization algorithms and their applications.
E-mail: csudtx@sina.com