International Journal of Computational Intelligence Systems, Vol.4, No. 1 (February, 2011).

Particle Swarm Optimization with Novel Processing Strategy and Its Application

Yuanxia Shen
School of Information Science and Technology, Southwest Jiaotong University, Chengdu 600031, China;
Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
Department of Computer Science, Chongqing University of Arts and Sciences, Chongqing 402160, China
E-mail: yuanxiashen@163.com

Guoyin Wang
Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
E-mail:wanggy@cqupt.edu.cn

Chunmei Tao
Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
E-mail:taocm@cqupt.edu.cn

Accepted: 01-12-2009
Received: 02-10-2010

Abstract

The loss of population diversity is one of main reasons which lead standard particle swarm optimization (SPSO) to suffer from the premature convergence when solving complex multimodal problems. In SPSO, the personal experience and sharing experience are processed with a completely random strategy. It is still an unsolved problem whether the completely random processing strategy is good for maintaining the population diversity. To study this problem, this paper presents a correlation PSO model in which a novel correlative strategy is used to process the personal experience and sharing experience. The relational expression between the correlation coefficient and population diversity is developed through theoretical analysis. It is found that the processing strategy with positive linear correlation is helpful to maintain the population diversity. Then a positive linear correlation PSO, PLCPSO, is proposed, where particles adopt the positive linear correlation strategy to process the personal experience and sharing experience. Finally, PLCPSO has been applied to solve single-objective and multi-objective optimization problems. The experimental results show that PLCPSO is a robust effective optimization method for complex optimization problems.

Keywords: Particle swarm optimization; correlation coefficient; population diversity, multi-objective optimization.

1. Introduction

Particle swarm optimization (PSO) is the swarm intelligent model inspired by certain social bird flocking and fish schooling [1]. Due to its simple operator and few parameters, PSO has been applied to solve real-world optimization problems successfully [2-5], including power system, image processing, economic dispatch, neural networks and some project applications, etc.

PSO emulates the swarm behavior and the individuals represent points in the search space. Assume a *D*-dimensional search space $S \subset R^D$ and a swarm consisting of *N* particles. The current position of the *i*-th particle is a *D*-dimensional vector $\boldsymbol{X}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]^{\mathrm{T}}$. The velocity of the *i*-th particle is also a *D*-dimensional vector $\boldsymbol{V}_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]^{\mathrm{T}}$. In every search-iteration, each particle is updated by following two “best” values, called P_i and P_g. P_i is the prior personal position of the *i*-th particle (also known as *pbest*), i.e. the personal experience. P_g is the best position found by particles in the swarm (also known as *gbest*), i.e. the sharing experience. The velocity V_{id} and position X_{id} of the *d*-th dimension of the *i*-th particle are updated with the following equations.

$$\begin{aligned} V_{id}(t+1) = wV_{id}(t) + c_1 r1_{id}(t)(P_{id}(t) - X_{id}(t)) \\ + c_2 r2_{id}(t)(P_{gd}(t) - X_{id}(t)) \end{aligned} \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

where random factors $r1_{id}$ and $r2_{id}$ are two independent random numbers in the range [0, 1]; *w* is an inertia weight; c_1 and c_2 are acceleration coefficients reflecting the weighting of stochastic acceleration terms that pull each particle toward *pbest* and *gbest* position, respectively. The first part of Eq. (1) represents the previous velocity, which provides the necessary momentum for particles to roam across the search space. The second part, known as the cognitive component, represents the natural tendency of individuals to return to environments where they experienced their best performance. The third part is known as the social component, which represents the tendency of individuals to follow the success of other individuals.

Although PSO has been applied to solve many optimization problems successfully, it may easily suffer from the premature convergence when solving complex problems. Many researchers have worked on improving the performance of PSO in various ways. To maintain the population diversity is a main objective of much work. Shi and Eberhart proposed a linearly decreasing inertia weight (LDIW) and a fuzzy adaptive inertia weight, which are used to balance the global and local search abilities [6-7]. To weaken the search density surrounding the historical best position found by the whole swarm in the early evolution, Asanga [8] developed the dynamic strategy that the cognitive coefficient decreases linearly from 2.5 to 0.5, while the social coefficient increases linearly from 0.5 to 2.5. Another active research trend in PSO is the design of different topological structures. Kennedy [9] claimed that PSO with a small neighborhood might perform better on complex problems, while PSO with a large neighborhood would perform better on simple problems. The ring topological structure and the von Neumann topological structure are proposed to restrict the information interaction among particles for relieving the loss of population diversity [10]. Suganthan [11] applied a dynamically adjusted neighborhood where the neighborhood of a particle gradually increases until it includes all particles. Parsopoulos and Vrahatis [12] combined the global version and local version together to construct a unified particle swarm optimizer (UPSO). Mendes et al. [13] proposed a fully informed particle swarm (FIPS), where the information of the entire neighborhood is used to guide the particles. To increase the population diversity, perturbation operator [14], evolution operator [15] and other search algorithms [16] are introduced to PSO. In addition, Xie and Zhang [17] presented a self-organizing PSO based on the dissipative system in which the negative entropy is introduced to improve the population diversity. Jie et al. [18] introduced a knowledge billboard to record varieties of search information and take advantage of multi-swarm to maintain the population diversity and guide their evolution by the shared information.

However, these PSO algorithms follow the same principle that each particle adopts the completely random strategy for processing the *pbest* and *gbest*, which lets the cognitive and the social components of the whole swarm contribute randomly to the position of each particle in the next iteration. Although the original objective of the completely random processing strategy is to keep the randomness of search, it is still an unsolved problem whether this strategy is good for maintaining the population diversity. To study this problem, we propose a correlation PSO model in which a novel correlative processing strategy is used to process the *pbest* and *gbest*. Then the relationship between the degree of correlation and population diversity is presented, which shows that the processing strategy with positive linear correlation has advantage of maintaining population diversity. In order to improve the global optimization ability of PSO, a positive linear correlation PSO (PLCPSO) is proposed in the context of the correlation PSO model.

Optimization plays a major role in the modern-day design and decision-making activities. Particularly, the multi-objective optimization (MOO) becomes a challenging problem due to the inherent confliction nature of objective to be optimized. As evolutionary algorithm (EA) can deal simultaneously with a set of possible solutions in a single run, it is especially suitable to solve MOO problems. Many evolutionary multi-objective optimization algorithms have been developed in the last few years, such as evolutionary computation, swarm intelligence [19-21]. As a new form of swarm intelligence, PSO has been used to solve MOO problems. To maintain the population diversity, several techniques [21-24] are introduced to PSO, e.g. an adaptive-grid mechanism, an adaptive mutation operation. In this paper, PLCPSO with the disturbance operation is used to solve MOO problems, where the correlative processing strategy is employed to maintain the population diversity.

The remainder of this paper is organized as follows. A positive linear correlation PSO is introduced in section 2. Simulation experiment results on some benchmark optimization problems are discussed in section 3. Conclusions are drawn in section 4.

2. PLCPSO

2.1. *The correlation PSO Model*

In PSO, each particle follows the *pbest* and *gbest* to search for a better position. Therefore, the strategy for processing the *pbest* and *gbest* influences the course of search. From the aspect of cognitive study, if a particle considers that the *pbest* is important for the search toward the optimum solution, then the *gbest* should be also important because the *gbest* is the best among all *pbest*s. Therefore, it should be discriminated to exploit the *pbest* and *gbest*. However, in SPSO, each particle adopts a completely random processing strategy, which makes no difference to exploit the *pbest* and *gbest*. Hence, it is concerned how to make proper use of the *pbest* and *gbest*. From Eq. (1), acceleration coefficients and random factors jointly affect the exploitation of *pbest* and *gbest*. Much work has focused on adjusting acceleration coefficients for balancing the global exploration and local exploitation, but the little attention is paid to random factors. In this paper, we are concerned about the effect of random factors on the performance of PSO. To study this problem, a correlation PSO model is presented, where random factors are correlated, and then particles adopt the correlative strategy to process *pbest* and *gbest*.

In this paper, we focus on the linear correlation between random factors. The correlation coefficient Spearman's $\rho_{X,Y}$ is a useful tool for measuring the strength of the linear correlation between random variables X and Y [25]. Thereby the Spearman's ρ is used to measure the correlation of random factors. The velocity of the *i*-th particle in the correlation PSO model is updated as follows:

$$\begin{cases} V_{id}(t+1) = wV_{id}(t) + c_1 r1_{id}(t)(P_{id}(t) - X_{id}(t)) \\ \qquad\qquad + c_2 r2_{id}(t)(P_{gd}(t) - X_{id}(t)) \\ \rho^{i}_{r_1,r_2}(t) = \alpha, (-1 \le \alpha \le 1) \end{cases} \tag{3}$$

where $\rho^{i}_{r_1,r_2}(t)$ is the correlation coefficient of the random factors of the *i*-th particle and *a* is a number in [0,1]. The position of each particle in the correlation PSO model is updated by Eq. (2). In Eq. (3), the different correlation coefficients can decide the different correlative strategies for processing the *pbest* and *gbest.* The range of the correlation coefficient is from -1 to 1. When *a*=0, the algorithm is the SPSO, where particles adopt the completely random strategy. SPSO is a special case of the correlation PSO model; When -1≤*a*<0, the algorithm is called the negative linear correlation PSO (NLCPSO), where the processing strategy with negative linear correlation denotes that particles enhance the exploitation of the one of the *pbest* and *gbest*, meanwhile weaken the exploitation of the other; If 0<*a*≤1, the algorithm is called the positive linear correlation PSO (PLCPSO), where the processing strategy with positive linear correlation denotes that particles enhance the exploitation of the one of the *pbest* and *gbest*, meanwhile enhance the exploitation of the other.

The correlation PSO model extends the information processing mechanism of PSO. The population diversity affects the performance of PSO greatly. For different processing strategies, it is concerned which kind of strategy can improve the global optimization ability of PSO. In the next section, the relationship between the population diversity and processing strategies should be analyzed.

2.2. *The Analysis of population diversity of PLCPSO*

The high population diversity should directly imply that a large area of the search space is being explored [26]. Similarly, the low population diversity should imply that the particles are exploiting a small area of the search space. In SPSO, the loss of population diversity is one of main reasons which lead SPSO to suffer from the premature convergence when solving complex multimodal problems. Therefore, maintaining population diversity is an important method for improving the global optimization ability of PSO. In this section, we discuss the relationship between the population diversity and correlation coefficient in the correlation PSO model.

Generally, the degree of dispersion of the particles in the swarm is used to measure the population diversity. This measure is given in [27] by:

$$div(X(t))=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}[X_{id}(t)-\overline{X_d(t)}]^2 \tag{4}$$

where N is the swarm size, D is the dimensionality of the problem and X_{id} is the d-th dimension of the i-th particle position. $\overline{X_d(t)}$ is calculated by the following equation.

$$\overline{X_d(t)}=\sum_{i=1}^{N}X_{id}(t)/N \tag{5}$$

In the correlation PSO model, particles can adopt different strategies to process *pbest* and *gbest*. In order to obtain the relationship between different correlative strategies and population diversity, the change of population diversity at the next time step is studied in the context of the current state. The population diversity at the time step t+1 can be represented by

$$div(X(t+1))=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}[X_{id}(t+1)-\overline{X_d(t+1)}]^2 \tag{6}$$

where the position and velocity of each particle at the time step t and before the time step t are known. The positions of particles at the time step t+1 are calculated by the Eqs. (2) and (3). Obviously, $X_{id}(t+1)$ and $\overline{X_d(t+1)}$ are random variables because $r1_{id}$ and $r2_{id}$ are random numbers. The population diversity $div(X(t+1))$ at the time step t+1 is also a random variable.

For the randomness of the population diversity at the time step t+1, we take the maximization method of expected utility to decide which is favorable to maintain population diversity among the correlative strategies. The expectation of $div(X(t+1))$ is calculated by

$$E[div(X(t+1))]=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}E[X_{id}(t+1)-\overline{X_d(t+1)}]^2$$

$$=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}E[X_{id}(t+1)-E(X_{id}(t+1))+E(X_{id}(t+1))-\overline{X_d(t+1)}]^2$$

$$=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{E[X_{id}(t+1)-EX_{id}(t+1)]^2-2E([X_{id}(t+1)-$$
$$EX_{id}(t+1)][\overline{X_d(t+1)}-EX_{id}(t+1)])+E[\overline{X_d(t+1)}-EX_{id}(t+1)]^2\}$$

$$=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{Var[X_{id}(t+1)]+\frac{1}{N}Var[X_{id}(t+1)]+\frac{1}{N^2}\sum_{j=1}^{N}Var[X_{jd}(t+1)]$$
$$+(\frac{1}{N}\sum_{j=1}^{N}E[X_{jd}(t+1)]-E[X_{id}(t+1)])^2\}$$

$$=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{(1-\frac{1}{N})Var[X_{id}(t+1)]\}+\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}(E[X_{id}(t+1)]$$
$$-\frac{1}{N}\sum_{j=1}^{N}E[X_{jd}(t+1)])^2$$

$$=\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{(1-\frac{1}{N})Var[X_{id}(t+1)]\}+div(E[X(t+1)]) \tag{7}$$

The expectation of $X_{id}(t+1)$ can be calculated as follow:

$$E(X_{id}(t+1))=X_{id}+wV_{id}(t)+0.5c_1(P_{id}(t)-X_{id}(t))$$
$$+0.5c_2(P_{gd}(t)-X_{id}(t)) \tag{8}$$

Then the variance of $X_{id}(t+1)$ can be obtained.

$$Var[X_{id}(t+1)]=\frac{1}{12}\{c_1^2[P_{id}(t)-X_{id}(t)]^2+2\rho c_1c_2[P_{id}(t)-$$
$$X_{id}(t)][P_{gd}(t)-X_{id}(t)]+c_2^2[P_{gd}(t)-X_{id}(t)]^2\} \tag{9}$$

Substituting Eq. (9) into Eq. (7), we get the expectation of $div(X(t+1))$.

$$E[div(X(t+1))]=E_0[div(X(t+1))]+\frac{\rho c_1c_2}{6ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{[P_{id}(t)-$$
$$X_{id}(t)][P_{gd}(t)-X_{id}(t)]\} \tag{10}$$

Where $E_0[div(X(t+1))]$ is the expectation of population diversity at the next time step t+1 in which the correlation coefficient is zero. Then $E_0[div(X(t+1))]$ is also the expectation of population diversity of SPSO at the next time step. $E_0[div(X(t+1))]$ is calculated by

$$E_0[div(X(t+1)]=\frac{N-1}{12N^2D}\sum_{i=1}^{N}\sum_{d=1}^{D}\{c_1^2[P_{id}(t)-X_{id}(t)]^2+c_2^2[P_{gd}(t)-$$
$$X_{id}(t)]^2\}+div(E[X(t+1)]) \tag{11}$$

To make clear the relationship between the correlation coefficient and population diversity, it is crucial to analyze the sign of the value of the second term in Eq. (10). The second term $\sum\sum[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]$ is the sum of ND products of $(P_{id}(t)-X_{id}(t))$ and $(P_{gd}(t)-X_{id}(t))$. According to the relations of the positions among $P_{id}(t)$, $X_{id}(t)$ and $P_{gd}(t)$, if $X_{id}(t)$ is not in the middle of $P_{gd}(t)$ and $P_{id}(t)$, then $[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]>0$. There exist the two possibilities, which is $X_{id}(t)$ lie either the left or right side of $P_{gd}(t)$ and $P_{id}(t)$. If $X_{id}(t)$ is in the middle of $P_{gd}(t)$ and $P_{id}(t)$, then $[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]<0$. Hence, the probability of the inequality $[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]>0$ is equal to 2/3, i.e. $\boldsymbol{P}([P_{id}(\mathrm{t})-X_{id}(\mathrm{t})][P_{gd}(\mathrm{t})-X_{id}(\mathrm{t})]>0)=2/3$. For all particles in the swarm, at least there exists the particle which the best personal position $P_{id}(t)$ should be equal to the best position $P_{gd}(t)$among all the particles in the swarm. Then, in this case, $[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]>0$, and then $\boldsymbol{P}([P_{id}(\mathrm{t})-X_{id}(\mathrm{t})][P_{gd}(\mathrm{t})-X_{id}(\mathrm{t})]>0)>2/3$.

Without loss of generality, assume $|[P_{id}(\mathrm{t})-X_{id}(\mathrm{t})][P_{gd}(\mathrm{t})-X_{id}(\mathrm{t})]|=\theta+\eta$, where θ is a positive constant, $\eta\sim N(0,\delta)$ is a white noise. Then $\mu=E[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]>0$.

According to the law of large numbers in probability theory, for any given number $\varepsilon>0$, the following equation can be obtained.

$$\lim_{ND\to\infty} P(\left|\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]\}-\mu\right|<\varepsilon)=1 \qquad (12)$$

Eq. (12) can deduce Eq. (13), thereby

$$\lim_{ND\to\infty} P(\frac{1}{ND}\sum_{i=1}^{N}\sum_{d=1}^{D}\{[P_{id}(t)-X_{id}(t)][P_{gd}(t)-X_{id}(t)]\}>0)=1 \qquad (13)$$

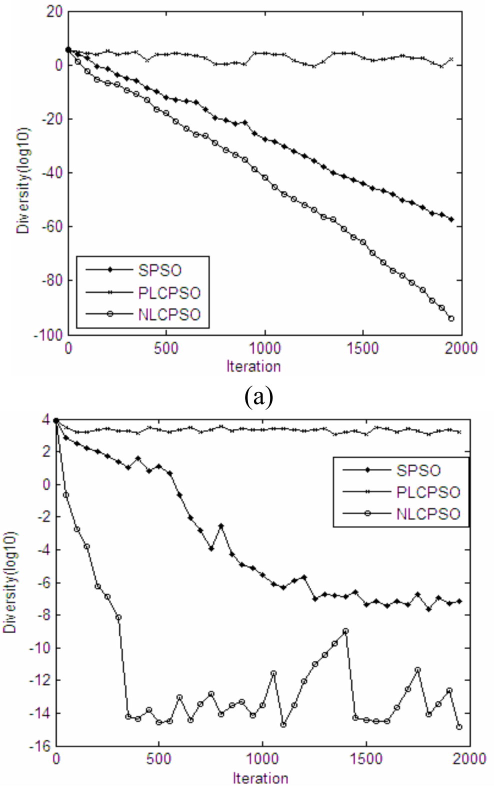
Synthesize Eq. (10) and (13), the conclusion is obtained that the expectation of the population diversity increases with the increasing of the correlation coefficient. The conclusion shows that PLCPSO is helpful for maintaining the population diversity, which makes particles in PLCPSO have more chance to get away from the local optima than SPSO and NLCPSO. From Eq. (10), it is obvious that the NLCPSO is easier to lose the population diversity than SPSO. When the correlation coefficient is set to be 1, the expectation of the population diversity can reach the maximum value. Likewise, when the correlation coefficient is set to be -1, the expectation of the population diversity can reach the minimum value. In this paper, we only consider two special cases, i.e. the correlation coefficient set to be -1 and 1. In the following contents, NLCPSO and PLCPSO specially denote the PSO algorithms in which the correlation coefficients are set to be -1 and 1, respectively.

In order to test the above analysis of the population diversity, PLCPSO, SPSO and NLCPSO are run 20 times on a (unimodal) sphere function and a (multimodal) Rastrigin function defined in Section 3. The changes of population diversity with iterations for each function are shown in Fig.1.

As can been seen from Fig.1, PLCPSO maintains the higher population diversity than SPSO and NLCPSO. The population diversity of SPSO and NLCPSO decrease with the increasing iteration, which makes SPSO and NLCPSO easily get trapped in local optima in later evolution. Meanwhile, the population diversity of SPSO decreases more slowly than NLCPSO. The experimental results are agreed with the analysis of the population diversity.

2.3. *Implementation of search velocity*

To enhance the speed of the search, a small modification is introduced to the velocity of the particle. If the velocity of a particle is zero, then the velocity of this particle is set to be a random number in the range from the lower bound to the upper bound of the velocity. The bounds of the velocity are specified by the user and applied to clamp the maximum velocity of each particle. Usually, the bounds of the velocity are set as the search bounds of the position.



(a)

(b)

Fig.1 Comparison of SPSO's, PLNPSO's and NLSPSO's population diversity. (a) Curve of population diversity for Shpere function. (b) Curve of population diversity for Rastrigin function

3. Applications

3.1. *Experiment 1*: *single-objective optimization*

In order to test the effectiveness of PLCPSO, six famous single-objective benchmark functions were optimized by PSO with linearly decreased inertial weight (PSO-LDIW), PSO with time-varying acceleration coefficients (PSO-TVAC), the fully informed particle swarm (FIPS), NLCPSO and PLCPSO.

3.1.1. *Test Functions*

The six benchmark functions include three unimodal functions (f_1~f_3) and three multimodal functions (f_4~f_6). The multimodal functions have complex multimodal distribution with one or multiple global optima enclosed by many local minimizations. All test functions have to be minimized. The properties and the formulas of functions are presented below.

Sphere's function

$$f_1(x)=\sum_{i=1}^{D}x_i^2\text{，}$$

$x\in[-100,100]$, D=30,min(f_1)=f_1(0,0,…,0)=0.

Quadric's function

$$f_2(x)=\sum_{i=1}^{D}(\sum_{k=1}^{i}x_k)^2$$

$x\in[-100,100]$, D=30,min(f_2)=f_2(0,0,…,0)=0.

Rosenbrock's function

$$f_3(x)=[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$$

$x\in[-10,10]$, D=30, min(f_3)=f_3(1,1,…,1)=0.

Rastrigin's function

$$f_4(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$$

$x\in[-5.12,5.12]$, D=30, min(f_4)=f_4(0,0,…,0)=0.

Noncontinuous Rastrigin's function

$$f_4(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi y_i)+10)$$

$$where \quad y_i=\begin{cases} x_i & |x_i|<0.5 \\ round(2x_i)/2 & |x_i|\geq 0.5 \end{cases}$$

$x\in[-5.12,5.12]$, D=30,min(f_5)=f_5(0,0,…,0)=0.

Shaffer's function

$$f_6(x)=\sum_{i=1}^{D-1}(x_i^2+x_{i+1}^2)^{0.25}[\sin(50(x_i^2+x_{i+1}^2)^{0.1}+1.0]$$

$x\in[-100,100]$, D=30, min(f_6)=f_6(0,0,…,0)=0.

3.1.2. *Parameters Setting for PSO Algorithms*

Parameters setting for PSO-LDIW, PSO-TVAC and FIPS come form Refs. [6], [8] and [13]. In FIPS, the ring topology structure is implemented with weighted FIPS for higher successful ratio, as recommended in [13]. In PSO-LDIW, FIPS, NLCPSO and PLCPSO, the inertia weight is decreased linearly from 0.9 to 0.4, and c_1= c_2=2. In PSO-TVAC, the cognitive coefficient decreases linearly from 2.5 to 0.5, while the social coefficient increases linearly from 0.5 to 2.5. For a fair comparison among all the PSO algorithms, they are tested using the same population size of 40. Further, the maximum fitness evaluation (FE) is set at 200000 for each test function. For the purpose of reducing statistical errors, each function is independently simulated 30 times, and their mean results are used in the comparison.

3.1.3. Experiment Results and Discussions

Table 1 presents the means and standard deviations of the 30 runs of the five algorithms on the six test functions. The best results among the five algorithms are shown in bold. Fig.2 presents the comparison in terms of the convergence characteristics of the evolutionary processes of each algorithm for each test function.

Table 2 Results of different PSO algorithms

Functions		PSO-LDIW	PSO-TVAC	FIPS	NLCPSO	PLCPSO
f_1	Mean	7.321×10^{-140}	6.628×10^{-81}	5.4662×10^{-6}	**2.243×10^{-191}**	8.371×10^{-101}
	Std. Dev	2.011×10^{-137}	9.538×10^{-79}	1.387×10^{-5}	**2.739×10^{-194}**	9.426×10^{-105}
f_2	Mean	3.455×10^{-42}	6.390×10^{-22}	7.882×10^{-3}	**4.990×10^{-119}**	1.982×10^{-13}
	Std. Dev	5.330×10^{-36}	9.816×10^{-20}	6.001×10^{-2}	**3.221×10^{-120}**	1.012×10^{-14}
f_3	Mean	29.82	13.84	21.99	**1.102**	2.239
	Std. Dev	20.53	18.77	23.32	5.628	**1.872×10^{-3}**
f_4	Mean	28.56	26.72	51.29	48.97	**1.414×10^{-9}**
	Std. Dev	17.35	24.93	22.87	10.76	**8.322×10^{-8}**
f_5	Mean	31.66	25.12	46.99	45.89	**3.947×10^{-5}**
	Std. Dev	11.57	13.81	21.13	10.12	**6.445×10^{-4}**
f_6	Mean	7.857	8.531×10^{-1}	25.91	2.379	**7.548×10^{-2}**
	Std. Dev	9.891	1.676	13.64	5.967	**2.126×10^{-3}**

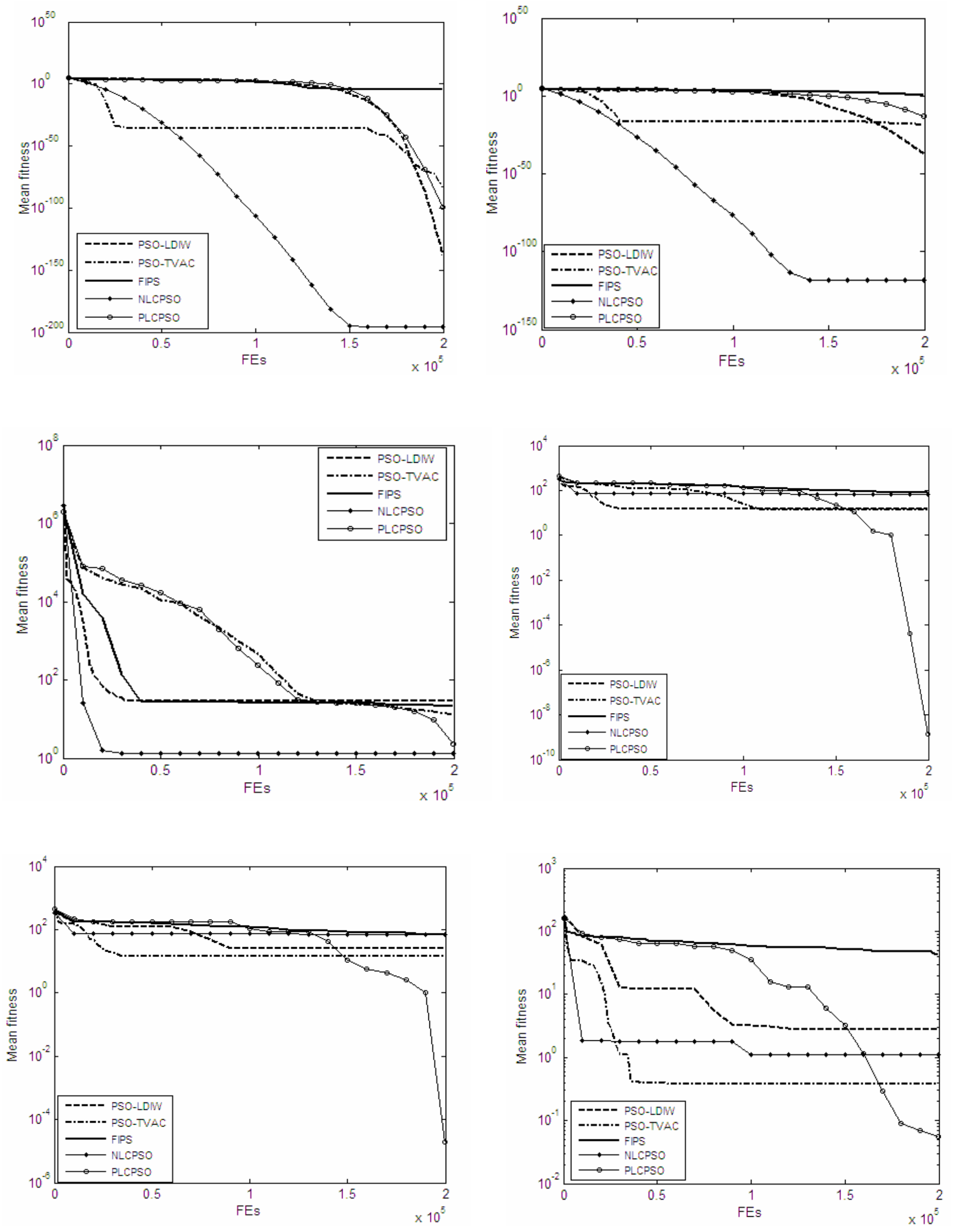


Fig.2 The convergence curve of test function for different function. (a) Sphere's function. (b) Quadric's function (c)Rosenbrock's function (d) Rastrigin's function (e) Noncontinuous Rastrigin's function (f) Schaffer's function

For unimodal functions, from the results, NLCPSO achieved the best means because the low population diversity enhances the local search ability of NLCPSO; PLCPSO has the good results, especially for the difficult Rosenbrock's function. For multimodal functions, PLCPSO surpasses all other algorithms, and avoids getting into the premature convergence, which benefits from the high population diversity. The experimental results show that the low population diversity is helpful for simple unimodal functions, while the high population diversity is good for complex multimodal problems.

Comparing the results and the convergence graphs, among these five algorithms, PSO-LDIW, PSO-TVAC and NLCPSO get trapped in the local optima for the difficult unimodal functions (e.g. Rosenbrock's function f_3) and the multimodal functions because of the rapid loss of the population diversity. FIPS with a ring topology has a local neighborhood, which can avoid falling into the premature convergence. However, the local neighborhood leads FIPS to converge slowly, and FIPS cannot achieve the satisfied results.

Since PLCPSO has the high population diversity, it could not converge as fast as NLCPSO for unimodal functions. Hence PLCPSO does not perform the best for simple unimodal functions. According to the theorem of "no free lunch" [28], one algorithm cannot offer better performance than all the others on every aspect or on every kind of problem. Therefore, we may not expect the best performance on all classes of problems, as the proposed PLCPSO focuses on improving the performance of PSO on complex multimodal problems.

3.2. *Experiment 2: multi-objective optimization (MOO)*

3.2.1. *Basic concepts on MOO*

In general, many real-world applications involve complex optimization problems with various competing specifications and constraints. Without loss of generality, we consider a minimization problem with decision space ***Y*** which is a subset of real numbers. For the minimization problem, it tends to find a parameter set y for

$$\underset{y\in Y}{Min}\, F(y) \quad y \in R^D \tag{14}$$

where $y = [y_1, y_2, \ldots, y_D]$ is a vector with D decision variables and $F = [f_1, f_2, \ldots, f_M]$ are M objectives to be minimized.

In the absence of any preference information, a set of solutions is obtained, where each solution is equally significant. The obtained set of solutions is called non-dominated or Pareto optimal set of solutions. Any solution $y = [y_1, y_2, \ldots, y_D]$ dominates $z = [z_1, z_2, \ldots, z_D]$ if and only if y is partially less than z, i.e., $\forall i \in 1, \cdots, D$

$$f_i(y) \le f_i(z) \wedge \exists i \in \{1, \cdots, D\} : f_i(y) < f_i(z) \tag{15}$$

The front obtained by mapping the Pareto optimal set (O_S) into the objective space is called POF

$$POF = \{\bar{f} = (f_1(x), \cdots, f_D(x)) \,|\, x \in O_S\} \tag{16}$$

The determination of a complete POF is a very difficult task, owing to the presence of a large number of suboptimal Pareto fronts. By considering the existing memory constraints, the determination of the complete Pareto front becomes infeasible and, thus, requires the solutions to be diverse covering its maximum possible regions.

3.2.2. *Performance metrics*

The knowledge of Pareto front of a problem provides an alternative for selection from a list of efficient solutions. It thus helps in taking decisions, and also, the knowledge gained can be used in situations where the requirements are continually changing. In order to provide a quantitative assessment for the performance of MO optimizer, two issues are taken into consideration, i. e. the convergence to the Pareto-optimal set and the maintenance of diversity in solutions of the Pareto-optimal set. In this paper, convergence metric $\boldsymbol{\gamma}$ [22] and diversity metric $\boldsymbol{\delta}$ [22] have as qualitative measures. Convergence metric is used to measure the extent of convergence of the obtained set of solutions. The smaller is the value of $\boldsymbol{\gamma}$, the better is the convergence toward the POF. Diversity metric is used to measure the spread of solutions lying on the POF. For the most widely and uniformly spread out set of non-dominated solutions, diversity metric $\boldsymbol{\delta}$ would be very small.

3.2.3. *Description of PLC-MOPSO*

This section describes PLCPSO to MOO problem, called PLC-MOPSO. The motivation is to attain better convergence to the Pareto-optimal front. In PSO, the term *gbest* represents the best solution obtained by the whole swarm. Often the conflicting nature of the multiple objectives involved in MOO problems makes the choice of a single optimum solution difficult. To resolve this problem, the concept of non-dominance is used and an archive of non-dominance solutions is maintained, from which a solution is picked up as the *gbest* in PLC-MOPSO. The historical archive stores non-dominance solutions to prevent the loss of good particles. The archive is updated at each cycle, e.g., if the candidate solution is not dominated by any members in the archive, it will be added to the archive. Likewise, any archive members dominated by this solution will be removed form the archive. To obtain more solutions, the disturbance operation was adopted for randomly selected non-dominance solutions in the archive. PLC-MOPSO is described in Fig.3.

3.2.4 *Benchmark problems and PLC-MOPSO performance*

```
/$Ns$: size of the swarm; $MaxIter$: maximum member of iterations; $d$: the dimensions of the search space./
(1) $t$ = 0, randomly initialize $S_0$; /$S_t$: swarm at iteration t /
  • initialize $x_{i,j}$, $i \in \{1, . . . ,Ns\}$ and $j \in \{1, . . . ,d\}$
  / $x_{i,j}$: the $j$-th coordinate of the $i$-th particle /
  • initialize $v_{i,j}$, $i \in \{1, . . . ,Ns\}$ and $j \in \{1, . . . ,d\}$
  /$v_{i,j}$: the velocity of $i$-th particle in $j$-th dimension /
  • $P_i \leftarrow x_i$, $i \in \{1, . . . ,Ns\}$ / $P_i$: the coordinate of the personal best of the $i$-th particle /
(2)Evaluate each of the particles in $S_0$.
(3)$A_0 \leftarrow non\_dominated(S_0)$ /returns the non-dominated solutions from the swarm; $A_t$: archive at iteration $t$ /
(4) for $t$ = 1 to $t = MaxIter$:,
  •   for $i$ = 1 to $i = Ns$ / update the swarm /
  / updating the velocity of each particle /
  •     $v_i = w\, v_i + c_1 r_1 (P_i - x_i) + c_2 r_2 (Regb - x_i)$
  •     $\rho_t$=1; $r_1$=$rand$( ); $r_2$= $r_1$
  / $Regb$ is a value that is randomly taken form the archive/
  /updating coordinates /
  •     $x_i = x_i + v_i$
(5) Evaluate each of the particles in $S_t$.
(6) /updating the archive /
$A_t \leftarrow non\_dominated(S_t)$.
(7)/disturbance operation to the randomly selected solutions in $A_t$/
  $A_t \leftarrow Selected\ (non\_dominated(S_t))(1+0.5*rand(\ ))$;
(8) END while
  Return $A_t$
```

Fig.3 Pseudo code of PLC-MOPSO

In the context of MOO, the benchmark problems must pose sufficient difficulty to impede searching for the Pareto optimal solutions.

In this paper, four benchmark problems are selected to test the performance of the proposed PLC-MOPSO. Many researchers such as the authors in [21], [23] and [24] have applied these problems to examine their proposed algorithms. The definition of these test functions is summarized in Table 2.

In this experiment, the maximum fitness evaluation (FE) is set at 10000. The population size is set at 100 for all problems.

Table 2 Definition of the MOO problems

Test problem	Definition
Schaffer's study(SCH)	*Minimize* F=($f_1(x)$,$f_2(x)$), where $f_1(x) = x^2$ $f_2(x) = (x-2)^2$ $x \in [-10^3, 10^3]$
Fonseca's and Fleming's study (FON)	*Minimize* F=($f_1(x)$,$f_2(x)$), where $f_1(x) = 1 - \exp(-\sum_{i=1}^{3} \left(x_i - 1/\sqrt{3}\right)^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^{3} \left(x_i + 1/\sqrt{3}\right)^2)$ $x_i \in [-4, 4], i = 1, 2, 3$
Poloni's study (POL)	*Minimize* F=($f_1(x)$,$f_2(x)$), where $f_1(x) = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2]$ $f_2(x) = [(x_1 + 3)^2 + (x_2 + 1)^2]$ $A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2$ $A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2$ $B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2$ $B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2$ $x_i \in [-\pi, \pi], i = 1, 2$
Kursawe's study (KUR)	*Minimize* F=($f_1(x)$,$f_2(x)$), where $f_1(x) = \sum_{i=1}^{2} [-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})]$ $f_2(x) = \sum_{i=1}^{3} [\lvert x_i \rvert^{0.8} + 5\sin(x_i^3)]$ $x_i \in [-5, 5], i = 1, 2, 3$

Table 3 Results of the convergence metric for test problems

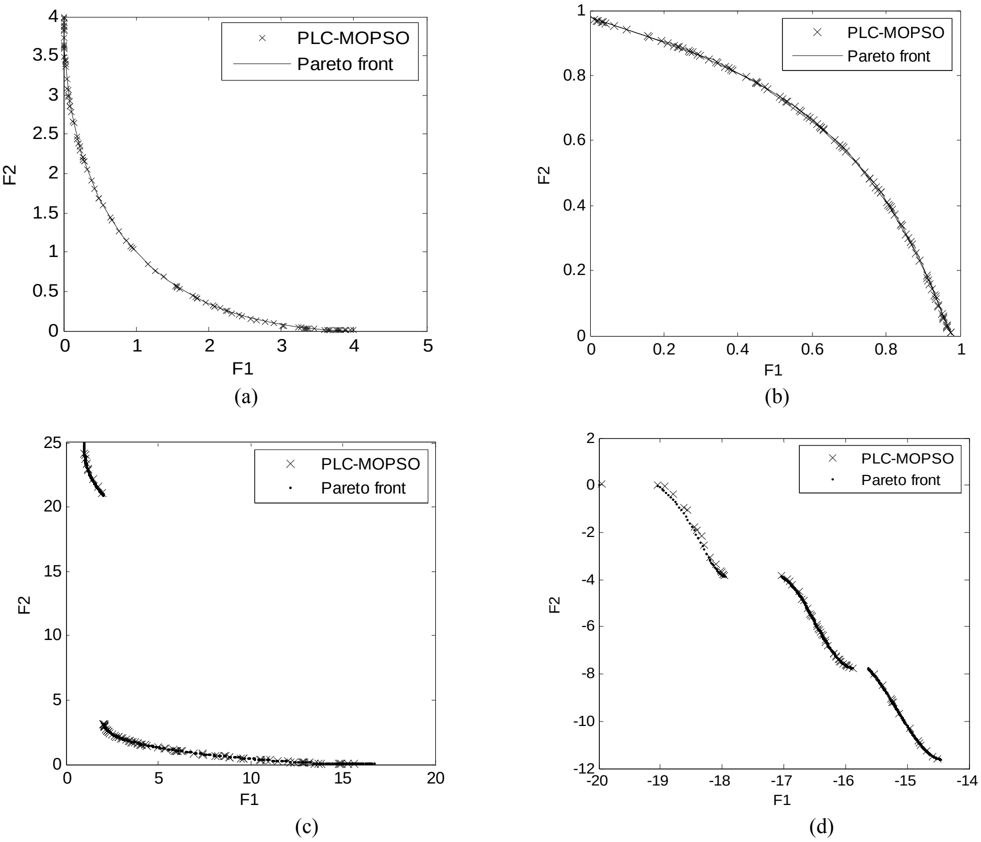
	ɣ	MOPSO	IPSO	PLC-MOPSO
SCH	Best	2.941e-3	**2.258e-3**	2.612e-3
	Mean	3.242e-3	2.638e-3	**2.567e-3**
	Worst	3.799e-3	3.430e-3	**3.031e-3**
	Std	4.900e-4	**4.410e-4**	4.562e-4
FON	Best	1.506e-03	1.378e-03	**1.234e-03**
	Mean	1.806e-03	1.517e-03	**1.448e-03**
	Worst	2.418e-03	2.437e-03	**2.123e-03**
	Std	1.100e-03	3.000e-04	**2.900e-04**
POL	Best	1.540e-02	**9.431e-03**	9.960e-03
	Mean	1.694e-02	**1.253e-02**	1.254e-02
	Worst	1.820e-02	1.343e-03	**1.270e-03**
	Std	2.300e-06	1.400e-06	**1.100e-06**
KUR	Best	2.136e-02	2.634e-02	**2.156e-02**
	Mean	2.647e-02	3.128e-02	**3.045e-02**
	Worst	3.242e-02	3.715e-02	**3.167e-02**
	Std	**2.700e-04**	4.500e-04	3.800e-04

Results for the convergence metric obtained using PLC-MOPSO, are given in Table 3, where results of MOPSO and IPSO come form Ref. [24]. From the results, they are evident that PLC-MOPSO converges better than MOPSO in all of test functions. For SCH and POL, the performance of PLC-MOPSO is very close to that of IPSO in converges metric. PLC-MOPSO converges slightly better than IPSO for FON and KUR. PLC-MOPSO and IPSO have the better diversity than MOPSO for SCH, POL and FON. However, for KUR, MOPSO has higher performance in diversity metric than PLC-

MOPSO and IPSO. In order to clearly visualize the quality of solutions obtained, figures have been plotted for the obtained Pareto fronts with POF. As can been seen form Fig. 4, the front obtained from PLC-MOPSO has the high extent of coverage and uniform diversity for all test problems. In a word, the performance of PLC-MOPSO is better than that of MOPSO, and is nearly close to that of IPSO in converges metric and diversity metric. It must be noted that MOPSO adopts an adaptive mutation operator and an adaptive-grid division strategy to improve its search potential, while IPSO adopts search methods including an adaptive-grid mechanism, a self-adaptive mutation operator, and a novel decision-making strategy to enhance balance between the exploration and exploitation capabilities. PLC-MOPSO only adopts disturbance operation to solve MOO problems, and no other parameters are introduced. This shows that the correlative strategy in PLC-MOPSO plays an important role in the global search for MOO problems.

Table 5 Results of the diversity metric for test problems

	δ	MOPSO	IPSO	PLC-MOPSO
SCH	Best	3.847e-01	3.385e-01	**3.134e-01**
	Mean	4.524e-01	**4.388e-01**	4.412e-01
	Worst	5.319e-01	5.189e-01	**4.819e-01**
	Std	3.570e-03	**3.430e-03**	3.541e-03
FON	Best	2.987e-01	**2.751e-01**	2.856e-01
	Mean	3.729e-01	3.162e-01	**3.098e-01**
	Worst	4.527e-01	3.794e-01	**3.527e-01**
	Std	8.500e-03	**1.140e-04**	9.800e-03
POL	Best	2.896e-01	2.962e-01	**2.755e-01**
	Mean	3.726e-01	3.140e-01	**3.041e-01**
	Worst	4.826e-01	3.419e-01	**3.154e-01**
	Std	2.435e-03	**1.980e-04**	2.000e-04
KUR	Best	3.725e-01	**3.927e-01**	3.913e-01
	Mean	**4.106e-01**	4.541e-01	4.408e-01
	Worst	**4.286e-01**	4.939e-01	4.912e-01
	Std	**8.470e-04**	1.200e-03	1.500e-03



(a)

(b)

(c)

(d)

Fig. 4 Nondominated solutions with PLCPSO for four MOO problems. (a)SCH, (b)FON, (c) POL, (d)KUR.

4. Conclusions

The contribution of this paper includes three parts. Firstly, a correlation PSO model is proposed, which extends information processing mechanism of particles and opens new avenues to improve the performance of PSO.

Secondly, the relational expression between the correlation of random factors and population diversity is presented through theoretical analysis. An important conclusion is obtained that the processing strategy with positive linear correlation is helpful for maintaining the population diversity. Thirdly, PLCPSO is presented in which particles adopt the positive linear correlation strategy to process the personal experience and sharing experience. This strategy is used to maintain the population diversity for improving the global search ability of PSO.

Experimental results show that PLCPSO is competitive in terms of performance, compared to the SPSO for the complex multimodal single-objective and multi-objective optimization problems. Another attractive property of PLCPSO is that it does not introduce any complex operations and new parameters to PSO framework. Thereby PLCPSO is also simple and easy to be implemented like the SPSO. Except for PLCPSO, it is interesting that NLCPSO surpasses other algorithms for unimodal functions though it may easily suffer from the premature convergence.

In order to improve the performance of PSO in complexly dynamic environment, our future research will be devoted to dynamic processing strategy.

Acknowledgements

This paper is supported by National Natural Science Foundation of China under Grant No.60773113; Natural Science Foundation of Chongqing under Grants No. 2008BA2017 and No.2008BA2041; Science & Technology Research Program of Chongqing Education Commission under Grant No.KJ090512; Chongqing Key Lab of Computer Network and Communication Technology No.CY-CNCL-2009-03.

References

1. J. Kennedy and R. C. Eberhart. Particle swarm optimization, in *Proceeding of International Conference on Neural Networks*. (Perth, Australia, 1995), pp. 1942-1948.
2. D. V. Yamille, K. V. Ganesh and M. Salman, Particle swarm optimization: basic concepts, variants and applications in power systems, *IEEE Transactions on Evolutionary Computation*. 12(2) (2008)171-195.
3. M. Madhubanti and C. Amitava, A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel threshold, *Expert Systems with Applications*. 34(2)(2008)1341-1350.
4. A. Mahor, V. Prasad and S. Rangnekar, Economic dispatch using particle swarm optimization: a review, *Renewable and Sustainable Energy Reviews*. 13(8)(2009)2134-2141.
5. J. B. Yu, S. J. Wang, L. F. Xi, Evolving artificial neural networks using an improved PSO and DPSO, *Neuron computing*. 71(4-6) (2008)1054-1060.
6. Y. H. Shi and R. C. Eberhart, A modified particle swarm optimizer, in *Proceeding of IEEE Congress on Evolutionary Computation*.(Piscataway, NJ, 1998), pp.69-73.
7. Y. H. Shi and R. C. Eberhart, Particle swarm optimization with fuzzy adaptive inertia weight, in *Proceeding of workshop particle swarm optimization*. (Indianapolis, IN, 2001), pp. 101–106.
8. R. Asanga, K. H. Saman and C. W. Harry, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE transaction on evolutionary computation*. 8(3) (2004) 240-255.
9. J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. in *Proceeding of IEEE Congress on Evolutionary Computation*. (Washington, DC, USA, 1999), pp.1931-1938.
10. J. Kennedy and R. Mendes, Population structure and particle swarm performance, in *Proceeding of IEEE Congress on Evolutionary Computation*. (Honolulu, HI, USA, 2002), pp. 1671-1676.
11. P. N. Suganthan, Particle swarm optimizer with neighborhood operator. *Proceeding of IEEE Congress on Evolutionary Computation*. (Washington, DC, USA, 1999), pp. 1958-1962.
12. K. E. Parsopoulos and M. N. Vrahatis, UPSO-a unified particle swarm optimization scheme, in *Proceedings of Computational Methods in Sciences and Engineering.* (Zeist, Netherlands, 2004), pp.868-873.
13. R. Mendes, J. Kennedy and J. Neves, The fully informed particle swarm: simpler, maybe better. *IEEE transaction on evolutionary computation*. 8(3) (2004) 204-210
14. X. C. Zhao. A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing*. 10(1) (2010)119-124
15. W. J. Zhang and X. F. Xie, Depso: hybrid particle swarm with differential evolution operator. in *Proceedings of IEEE international conference on systems, man and cybernetics*, (Washington, DC, USA, 2003),pp.3816-3821.
16. W. B. Langdon and P. Riccardo, Evolving problems to learn about particle swarm optimizers and other search algorithms, *IEEE transaction on evolutionary computation*. 11(5)(2007) 561-579.
17. X. F. Xie, W. J. Zhang and D. C. Bi. Optimizing semiconductor devices by self-organizing particle swarm. in

Proceeding of IEEE Congress on Evolutionary Computation, (Oregon, USA, 2004), pp. 2017-2022.

18. J. Jie, J. C. Zeng, C. Z. Han and Q. H. Wang. Knowledge-based cooperative particle swarm optimization. *Applied mathematics and computation*. 2(205)(2008)861-873.
19. Y. B. Liu and J. Huang, A novel fast multi-objective evolutionary algorithm for QoS multicast routing in MANET, International Journal of Computational Intelligence Systems 2-3(2009)288- 297
20. S. K. Chaharsooghi and A. H. M Kermani, An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP), Applied mathematics and computation, 200(1)(2008) 167-177
21. C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Transactions on evolutionary computation*. 3(3)(2004)256-280.
22. P. K. Tripathi, S. Bandyopadhyay and S. K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Information sciences*. 177(22) (2007) 5033–5049
23. D. S. Liu, K. C. Tan, C. K. Goh and W. K. Ho. A multi-objective memetic algorithm based on particle swarm optimization, *IEEE transaction on Systems, Man and Cybernetics, part b: Cybernetics*. 37(1)(2007)42-61
24. S. Agrawal, Y. Dashora, M. K. Tiwari and Y. J. Son, Interactive particle swarm: a pareto-adaptive metaheuristic to multiobjective optimization, *IEEE transaction on systems, man and cybernetics, part a: systems and humans*. 38(2) (2008) 258-278.
25. J. A. Rice (eds.), *Mathematical statistics and data analysis*, 2nd edn. (Thomson learning ,Wadsworth, 2004)
26. O. Olorunda and A. P. Engelbrecht. Measuring exploration/exploitation in particle swarms using swarm diversity, in *Proceeding of IEEE Congress on Evolutionary Computation*, (Hong Kong, China, 2008), pp. 1128-1134.
27. Y. H. Shi and R. C. Eberhart. Population diversity of particle swarm, *in Proceeding of IEEE Congress on Evolutionary Computation*. (Hong Kong, China, 2008), pp.1063-1068.
28. D. H. Wopert and W. G. Macreasy. No free lunch theorems for optimization, *IEEE Transactions on evolutionary computation*. 1(1)(1997)67-82.