Cooperative Particle Swarm Optimization Algorithm Base on Fuzzy Migratory Operator

Guo-Rong Cai, Yun-Dong Wu*, Shui-Li Chen
School of Science, Jimei University,
Research Centre of Image Information Engineering, Jimei University
Xiamen, Fjian Province, China, 361021

***Abstract*—In this article, a cooperative particle swarm optimization is proposed to increase the diversity of the swarm to avoid the drawback of trapping in a local optimum. Besides, a migratory operator on the basis of comprehensive fuzzy evaluation is employed to control the distribution of particles during iterations. The experimental results conducted on three benchmark functions show the performances of the proposed method are superior to that of classical PSO and particle swarm cooperative optimizer.**

Keywords-component; particle swarm optimization; migratory operator; population entropy; comprehensive fuzzy evaluation

I. Introduction

Particle swarm optimization (PSO) [1] is a population based global optimization method based on a simple simulation of bird flocking or fish schooling behavior. In PSO, the search points are known as particles, each particle is assigned a randomized velocity and is iteratively moved through the problem space. It is attracted towards the location of the best fitness achieved so far by the particle itself (pbest) and the location of the best fitness achieved so far across the whole population (gbest).

Although PSO can show significant performance in the initial iterations, the algorithm might encounter problems of local optimization in the latter iterations [2]. It is pointed out that the update of velocity of each particle works for much of exploitation and less of exploration. Consequently, the diversity of the population decreases sharply in the latter iterations so that the search may results in premature convergence. In order to prevent local optimal, some improved approaches including linear and nonlinear strategy of inertia setting, center PSO, landscape adaptive PSO and other variants of PSO have been reported[3~6]. In Ref. [7, 8, 9], the population is partitioned into several sub-swarms, each of which is made to evolve independently. According to the co-evolutionary relationship between swarms, diversity of population can be preserved during iterations. Ref. [10] introduced a diversity control method on the basis of information entropy to keep good balance between the exploration and the exploitation, thus reducing the risk of converging to local sub-optima.

This paper presents a two-layer cooperative particle swarm optimization to increase the diversity of particles. Furthermore, a fuzzy migratory operator is employed to induce particles search in local space. The performance of the proposed PSO is compared with some other available PSO models through analyzing the convergence speed, optimal results and variety of diversity.

II. Rules for Manipulation in PSO

As a population-based evolutionary algorithm, PSO is initialized with a population of candidate solutions and the activities of the population are guided by some behavior rules. For example, let $X_i(t)=(x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t))$, ($x_{id}(t)\in[-x_{d\max}, x_{d\max}]$) be the location of the ith particle in the tth generation, where $x_{d\max}$ is the boundary of the dth search space for a given problem. The location of the best fitness achieved so far by the ith particle is denoted as $p_i(t)$ and the index of the global best fitness by the whole population, as $p_g(t)$. The velocity of the ith particle is $V_i(t)=(v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t))$, where v_{id} is in $[-v_{d\max}, v_{d\max}]$ and $v_{d\max}$ is the maximal speed of the dth dimension. The velocity and position update equations of the ith particle are given as follows:

$$v_{id}(t+1) = w\cdot v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \tag{1}$$

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

where i=1,. . . , n and d=1,. . . , D. w, c_1, $c_2 \geq 0$. w is the inertia weight, c_1 and c_2 the acceleration coefficients, r_1 and r_2 are random numbers, generated uniformly in the range [0, 1], responsible for providing randomness to the flight of the swarm. The term $c_1r_1(p_{id} - x_{id})$ in Eq.(1) is called the cognition behavior and $c_2r_2(p_{gd} - x_{id})$ denoted the social behavior.

III. Fuzzy Migratory Strategy of Multi-Swarm Cooperative PSO

It is well known that a suitable distribution of particles during evolution will provides a balance between the global and local exploration ability of the swarm. In this section, a two-layer framework of PSO is proposed to increase the diversity of the population. In additional, a novel evaluate measure base on information entropy and comprehensive fuzzy evaluation is introduced to determine the diversity of swarm. Consequently, a migratory operator will be imposed on swarms which distributions are inappropriate according to the risk probability computed by comprehensive fuzzy evaluation.

The project is supported by the National Natural Science Foundation of Fujian Province, PRC (2008J0024); The key project of education department of Fujian province, PRC(JK2009017); The key project of institutes serving for the economic zone on the western coast of the Taiwan strait (2009(8)); The National Natural Science Foundation of Xiamen City, PRC (3502Z2093018).

A. The two-layer framework of multi-swarm cooperative PSO

In this article, a multi-swarm cooperative scheme, which consists of one swarm in the top layer and several swarms in the bottom layer is introduced. The bottom swarms can supply many new promising particles (the position giving the best fitness value) to the top swarm as the evolution proceeds. The top swarm updates the particle states based on the best position discovered so far by all the particles both in the bottom and its own.

In the initial population, M swarms of N particles, x^{jk}, j = 1, 2, . . . , M, k = 1, 2, . . . , N, are randomly generated in the bottom layer, where x^{jk} is the position of the kth particle in the jth swarm of the bottom layer. The update of x^{jk} depends on x^{ik}_{pbest} and x^{i}_{gbest}, where x^{ik}_{pbest} and x^{i}_{gbest} are the personal best position of x^{jk} and the global best position of the ith swarm, respectively. On the other hand, N particles are generated in the top swarm for pursing the global best solution.

B. Evaluation of diversity

In order to evaluate the diversity of a population, a time vary information entropy method is proposed. In our entropy base approach, the range of fitness value is divided into K even intervals according to best and the worst value of the swarm in the tth iteration. The definition of population entropy is given as follow:

Definition 1 Let $g_b{}^i(t)$ and $g_w{}^i(t)$ be the global best and the global worst value in the tth iteration of the ith swarm, respectively. Divide the interval I=[$g_b{}^i(t)$, $g_w{}^i(t)$] into K mean subintervals. Consequently, the population entropy of the ith swarm can be determined as follow:

$$D_i = -\sum_{j=1}^{K} \frac{n_j}{n} \lg(\frac{n_j}{n}) \tag{3}$$

where n and n_j are the population of the ith swarm and the number of particles in the jth interval.

Obviously, D_i=0 if and only if all particles of the ith swarm in a same interval. On the contrary, D_i=lg(n) if each particle in different interval. Thus $D \in [0,\ \lg(n)]$.

C. Migratory operator of PSO base on comprehensive fuzzy evaluation

It was inferred that the system should start with a high diversity for global exploration and the entropy value should slowly decrease in the later iterations rather than sharply cut down. Motivated by this, a fuzzy evaluation method can be employed to control the variety of the entropy.

1) The principles of comprehensive fuzzy evaluation of population entropy in PSO

The basic ideas of comprehensive fuzzy evaluation of population entropy are as follows: (1) Fuzzify population entropy and average fitness value of each sub-swarm according to the evaluation criteria; (2) adopt fuzzy sets transform principle to construct the fuzzy evaluation matrix on the basis of determining the rating criteria and the weights of population entropy and average fitness value; (3) determine their ranks through single-level computing. The main procedure is as follows:

- Let entropy and average fitness value of each sub-swarm be the factor set.
- Determine U={very low, low, medium, high, very high} as a set of reviews on entropy and average fitness value, where low and high etc are triangle fuzzy sets defined in entropy and average fitness value of swarms.
- Evaluate single factor to obtain fuzzy evaluation matrix.

  Expressed as follows:

  $$R = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} \\ r_{21} & r_{22} & r_{23} & r_{24} & r_{25} \end{pmatrix}$$

  where r_{ij} is membership degree, which can be obtained by selecting appropriate membership function of U.
- Single-level comprehensive evaluation.

  Induce the comprehensive fuzzy evaluation vector B via $B=W*R$, where W is the weight vector of the two factor and W=(0.5, 0.5).
- Convert evaluation vector into risk probability.

  An arithmetic sequence U=(0, 25, 50, 75, 100) is used to convert comprehensive reviews into risk probability via $P=U*B^T$, where P denotes the risk probability of the particle swarm
- Convert evaluation vector into risk probability.

  An arithmetic sequence U=(0, 25, 50, 75, 100) is used to convert comprehensive reviews into risk probability via $P=U*B^T$, where P denotes the risk probability of the particle swarm.

2) Applying fuzzy evaluation to migratory operator

Base on the slowly decrease purpose of population entropy, a migratory operator will be imposed on bottom swarms to control the distribution of particles according to the risk probability. So to speak, if the risk probability P is higher than a threshold of response ($\theta(t)$) of one bottom swarm, namely, the distribution of the swarm is inappropriate, a migratory operator will be employed to randomly choose $\sigma(t)$ particles via Eq.(5) to replace bad particles in the swarm.

$$\theta(t) = \alpha * (1 - \frac{t}{t_{\max}}) \tag{4}$$

$$\sigma(t) = \beta * n(1 - \frac{t}{t_{\max}}) \tag{5}$$

where $\alpha, \beta \in R$ are the migratory parameters and $t_{\max}$ denotes the maximum iteration.

IV. ALGORITHM

For each sub-swarm, do the following steps:

Step 1: Initialize the population of particles (randomly initialize n particles);

Step 2: Create population: Randomize the positions and velocities for entire population. Record the global best location p_g and the local best locations p_i of the ith particle.

Step 3: Evaluate the fitness value ($f(x_i)$) of ith particle through their positions, if ($f(x_i)$) < ($f(p_i)$), set p_i=x_i as the so far best position of the ith particle. If ($f(x_i)$) < ($f(p_g)$), set p_g=x_i as the so far best position of the population;

Step 4: Calculate the average fitness value and the entropy according to Eq.(3).

Step 5: Compute the risk probability θ according to the principles of comprehensive fuzzy evaluation.

Step 6: If swarm in the bottom layer has inappropriate risk probability ($P>\theta$), migratory operator will be imposed on the swarm to replace bad particles.

Step 7: Update the position and the velocity of all particles according to Eq.(1) and Eq.(2) (i=1,2, …, n);

Step 8: Termination criterion: Repeat Step3 to Step8 until Max number of generation or best solution.

V. NUMERICAL EXPERIMENTS

In this article, four versions of PSO are tested for three benchmark problems through analyzing the optimal results and the variety of population entropy. As a matter of convenience, the conventional, multi-swarm cooperative structure[7], adaptive entropy method[10] and the proposed PSO are denoted as method 'A', 'B', 'C', 'D', respectively. The results obtained through Method A, B, C and D are presented in Table 2.

A. Benchmark functions

The three non-linear benchmark functions selected in Table 1 are very often encountered in optimization algorithm benchmarks. f_1 is the Rosenbrock function whose global minimum is 0 at the point $x_i = 0$, $i = 1,\ldots,n$; The Rastrigin function, f_2, has lattice shaped semi-optimum solutions around the global optima, and there is no correlation among design variables. f_3 is the Griewank function who has a $\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})$ component causing linkages among variables, thereby making it difficult to reach the global optimum $x_i = 0$, $i = 1,\ldots,n$.

TABLE I. THREE BENCHMARK FUNCTIONS

Function	Equation	Dim.	Range	Vdmax	Goal
Rosenbrock	$f_1(x)=\sum_{i=1}^{D}(100\times(x_{i+1}-x_i^2)^2+(x_i-1)^2)$	10~30	$(-100，100)^d$	100	200
Rastrigrin	$f_2(x)=\sum_{i=1}^{D}(x_i^2-10\times\cos(2\pi x_i)+10)$	10~30	$(-10，10)^d$	10	100
Griewank	$f_3(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	10~30	$(-600，600)^d$	600	0.1

B. Parameter settings

Given the considerations mentioned above, our detailed parameter settings are as follows:

(1) $c_1 = c_2 = 2$, U=(0, 25, 50, 75, 100), α=0.8, β=0.5, W=(0.5, 0.5);

(2) The number of iterations is set as 1000 and algorithms are implemented for 100 runs. The results reported are on the basis of the average and the best performance obtained in each simulation.

(3) The inertia weight is decreased non-linearly from 0.95 to 0.4, that is, w_{initia}l=0.95, w_{final}=0.4[5];

(4) The ranges of the fitness functions, V_{dmax} and the goal of PSO are shown in Table 1.

(5) Let $a=\frac{f_w(t)-f_b(t)}{4}$ and $b=\frac{\lg(n)}{4}$, then the central points of triangle fuzzy membership functions defined in the average fitness value is $\{0,a,2a,3a,4a\}$,where $f_w(t)$ are $f_b(t)$ are the worst and the best average fitness value of bottom swarms in the tth iteration, respectively. On the other hand, central points of fuzzy membership functions defined in entropy of each swarm is $\{0,b,2b,3b,4b\}$.

(6) In method A and C, the population n=40, similarly, n=10 and M=4 for each swarm in method B and D..

C. Experiment results and discussions

The analyses are taken with four statistical parameters (average, best, standard deviation, the percentage of success of the test and the variety of population entropy) for every 1000 generations. All the four methods are simulated 100 times and the average results are recorded. From the results listed in Table2, it is observed that the proposed methods achieved the highest success rate of searching over 100 simulated runs. Besides, the average fitness values point out the proposed method is much better than the existing method A, B and C. The measurement of average deviation provides the information about the deviation of the fitness value obtained in each simulation run, from the average value of the fitness value which is taken over 100 simulations. Smaller value of SD indicates the consistency of the (fitness) optimal solution obtained in each simulation. As seen in Table 2, the proposed method produces better SD than existing methods.

On the other hand, the convergence analysis of the benchmark functions over 1000 generations for the existing and the proposed methods is plotted in Fig. 1 to Fig. 3. The results depicts that in almost all of the functions, the proposed model gives the fastest convergence toward the minimum and can find the global minima successfully. Furthermore, Fig 4 to Fig 6 depicts the iterations of population entropy obtained by conventional and the proposed PSO. It's convincing that the migratory operator can effectively maintains higher diversity of the swarm in the initial iterations which will enable particles to search in the solution space more aggressively and will effectively enhance the exploration in the later stage.

To sum up, it's evident the proposed method reflects the cooperative relationship among the top and the bottom swarms, the two-layer structure and the migratory mechanism helps the algorithm to search the solution space more aggressively to look for "better areas".

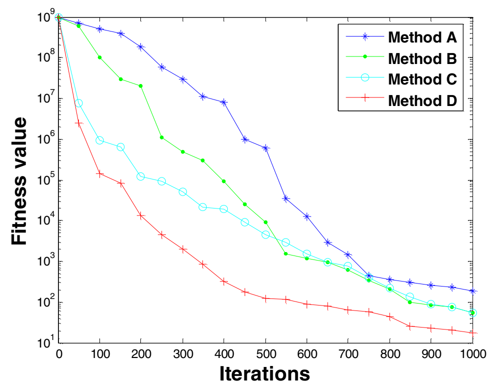

Figure 1. Iterations of fitness value of Rosenbrock function.

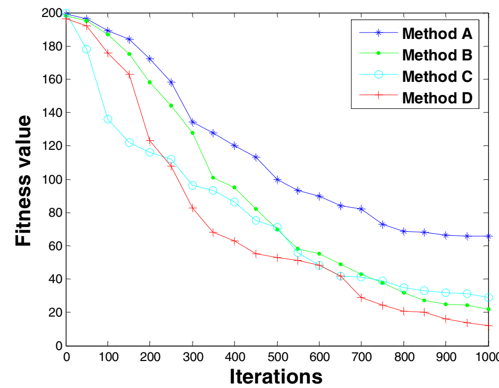

Figure 2. Iterations of fitness value of Rastrigrin function

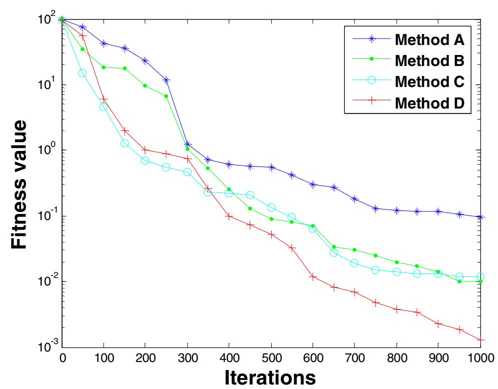

Figure 3. Iterations of fitness value of Griewank function.

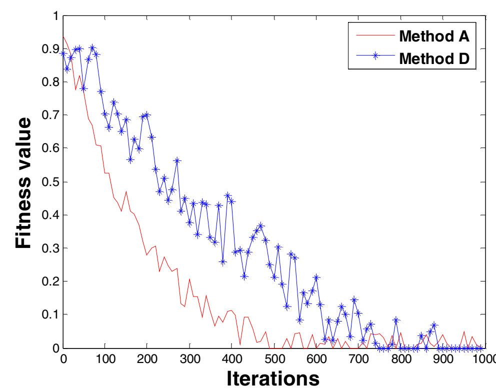

Figure 4. Itertions of population entropy of Rosenbrock function

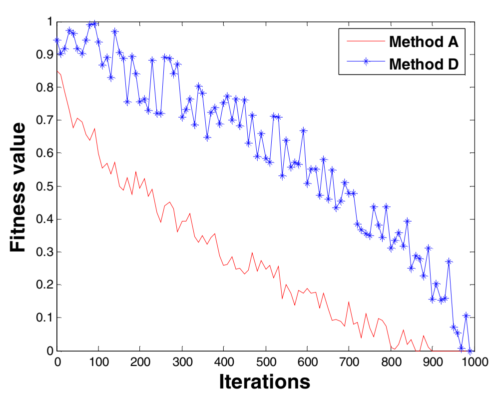

Figure 5. Itertions of population entropy of Rastrigrin function.

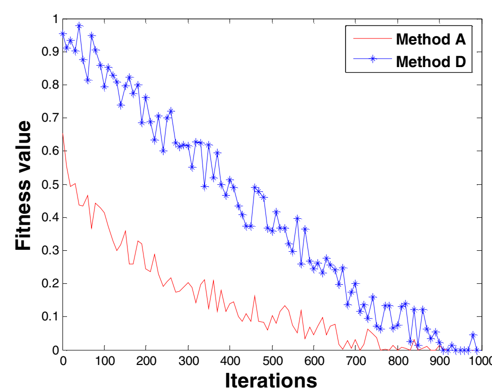

Figure 6. Itertions of population entropy of Griewank function.

TABLE II. COMPARISON OF THE FITNESS VALUES ACHIEVED BY FOUR AVAILABLE PSO MODELS

Function	Fitness	Method A	Method B	Method C	Method D
Rosenbrock	Best	4.368	0.1052	6.2564	3.2208
	Mean	190.798	54.2183	54.5129	17.54456
	SD	89.361	30.5751	33.0864	8.3622
	Suc.	78	92	94	100
Rastrigrin	Best	8.9963	10.2504	11.0625	4.6522
	Mean	66.3395	22.701	29.2358	12.1714
	SD	23.56	13.869	16.574	7.3698
	Suc.	82	87	93	99
Griewank	Best	5.73e-018	0	0	0
	Mean	0.095758	0.012	0.01564	0.0013
	SD	0.018662	0.011302	0.01056	0.0075
	Suc.	89	97	100	100

VI. CONCLUSION

A two-layer PSO base on fuzzy migratory operator is proposed in this paper. The interplay mechanism of swarms enable particles explore in local space during initial iterations and apply mild, fine tuning during later iterations so that the optimum solution can be approached with better accuracy. Besides, the performances of the proposed PSO algorithm have been tested for three benchmark functions and they have been compared with a number of popular PSO algorithms. The results are found to be mostly encouraging.

***Corresponding author**: Prof. Yun-Dong Wu.

REFERENCES

[1] Kennedy J, Eberhart R, Particle Swarm Optimization [A], International Conference on Neural Networks, IEEE, Perth, Australia, 1995, pp.1942-1948.

[2] Angeline PJ, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, Evolutionary Programming VII, Lecture Notes in Computer Science, Springer, Berlin , 1998. vol. 1447, pp. 601–610.

[3] Shi Y, Eberhart RC, A modified particle swarm optimizer, In: Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, NJ: 1998. pp. 69–73.

[4] Chen shuili, Cai Guo-Rong, etc, Study on the Nonlinear Strategy of Acceleration Coefficient in Particle Swarm Optimization Algorithm, Journal of Changjiang University, Jingzhou, China, 2007,4(4), 1-4.

[5] Cai Guo-R, Chen Shui-Li, etc, Study on the Nonlinear Strategies of Inertia Weight in Particle Swarm Optimization. Proceedings of the IEEE Congress on Natural Computing. 2008. 886-891.

[6] Jin Yisu, Joshua Knowles, etc, The landscape adaptive particle swarm optimizer, Applied Soft Computing, Elsevier, 2008, 8, pp.295-304.

[7] Li Ai-Guo, Particle Swarms Cooperative Optimizer, Journal of Fudan University, 2004 ,43 (5) :923 - 925.

[8] Chia-Chong Chen, Two-layer particle swarm optimization for unconstrained optimization problems, Applied Soft Computing, 2009, in press.

[9] Ben Niu, Yunlong Zhu ,etc, MCPSO: A multi-swarm cooperative particle swarm optimizer, Applied Mathematics and Computation, 2007 (185) 1050−1062.

[10] Duan Xiao-Dong, Gao Hong-Xia, etc, Adaptive Particle Swarm Optimization Algorithm Based on Population Entropy, Computer Engineering, 2007,33（18）:222-248.