Particle Swarm Optimisation from lbest to gbest

Hongbo Liu, Xiukun Wang

Abstract **The effects of various neighborhood models on the particle swarm algorithm were systematically investigated in this paper. We also gave some additional insight into the PSO neighborhood model selection topic. Our experiment results testified that gbest model converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while lbest model converges slowly on problem solutions but is able to "flow around" local optima, as the individuals explore different regions. The gbest model is recommended strongly for unimodal objective functions, while a variable neighborhood model is recommended for multimodal objective functions.**

***Index Terms*—Swarm Intelligence, Particle Swarm Optimization, Neighborhood Model, Global Optimum**

I. Introduction

Particle swarm optimization (PSO) was originally introduced by J. Kennedy et al. in 1995 as an optimisation technique inspired by swarm intelligence and theory in general such as bird flocking, fish schooling and even human social behavior [1]. Furthermore, the whole idea and structure of the algorithm is inspired by evolutionary computation. Later PSOs has turned out to be a worthy alternative to the standard genetic algorithm and other iterative optimisation techniques [2].The advantage of the PSO over many of the other optimization algorithms is its relative simplicity. The PSO algorithm is initialized with a population of random candidate solutions, conceptualized 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 and by the location of the best fitness achieved so far across the whole population (global version of the algorithm). The two most commonly used methods are known as gbest model and lbest model in particle swarm optimization. The gbest model converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while lbest model is able to "flow around" local optima, as the individuals explore different regions [3]. The lore is based on experience and some data, but population topologies have not been systematically explored. Sugarthan[4], Kennedy[5], Peer[6], Carlisle[7] researched effects of neighborhood models in different perspective, but some divergence exited among those results. And Kennedy and his colleague's research manipulated some sociometric variables that are hypothesized to affect performance. They discovered that previous assumptions may not have been correct [8]. The effects of various neighborhood models on the particle swarm algorithm are systematically investigated in this paper. The present work gives some additional insight into the PSO neighbourhood model selection topic.

The structure of this paper is as follows. Particle swarm optimization is overviewed briefly in Section Ⅱ. At the next section, we discuss the neighbourhood model of PSO algorithm. Then, the algorithm from lbest model and gbest model is described in detail in Section Ⅳ. Four main benchmark functions used as objective functions are analyzed in Section Ⅴ. Experiments, results and discussion are in Section Ⅵ and Section Ⅶ. Finally, we draw a conclusion about our work in Section Ⅷ.

II. Particle Swarm Optimization Algorithm

The basic PSO model consists of a swarm of particles moving in an n-dimensional search space where a certain quality measure, the fitness, can be calculated. Each particle has a position represented by a position-vector x and a velocity represented by a velocity-vector v. Each particle remembers its own best position so far in a vector p_i. i is the index of the particle and the d-th dimensional value of the vector p_i is p_{id} (i.e. the position where it achieved its best fitness). Further, a neighborhood relation is defined for the swarm. The best position-vector among all the neighbors of a particle is then stored in the particle as a vector p_g and the d-th dimensional value of the vector p_g is p_{gd}. At each iteration step the velocity is updated and the particle is moved to a new position. Firstly the update of the velocity from the previous velocity to the new velocity is (in its simplest form) determined by (1). And then the new position is determined by the sum of the previous position and the new velocity by (2), a neighborhood relation is defined for the swarm:

$$v_{id} = w * v_{id} + c_1 Rand() * (p_{id} - x_{id}) + c_2 Rand() * (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

In PSO algorithms, a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful neighbor. At each iteration, the particle with the best fitness in the local neighborhood, designated g, and the current particle

---

Manuscript received March 28, 2004. This work was supported by the MOST (2100CCA00700) and MOE (KP0302).

H. Liu is with Department of Computer, Dalian University of Technology, Dalian, 116023, China (phone: 0411-84706003; fax: 0411-82491643; e-mail: abrahamliu09@yahoo.com).

X. Wang is with Department of Computer, Dalian University of Technology, Dalian, 116023, China (e-mail: jsjwxk@dlut.edu.cn).

are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position is considered the *cognition* component, and the portion influenced by the best in the neighborhood is the *social* component. Later some actions differ from one variant of PSO to the other for improving the computational performance [9].

III. GBEST MODEL AND LBEST MODEL

Particle swarm algorithm can be described generally as a population of vectors whose trajectories oscillate around a region which is defined by each individual's previous best success and the success of some other particle, as above mention. Various methods have been used to identify "some other particle" to influence the individual. Eberhart and Kennedy called the two basic methods as "gbest model" and "lbest model" [3]. In lbest model, particles have information only of their own and their nearest array neighbours' best (lbests), rather than that of the entire group. Namely, in (1), gbest is replaced by lbests in the model. In the gbest population, the trajectory of each particle's search is influenced by the best point found by any member of the entire population. The best point/particle acts as an attractor, pulling all the particles towards it. Eventually all particles will converge to this position. The lbest population allows each individual to be influenced by some smaller number of adjacent members of the population array. The particles selected to be in one subset of the swarm have no relationship to the other particles in the other neighborhood. Typically lbest neighborhoods comprise exactly two neighbors. When the number of neighbors increasing to all but itself, the case is equivalent to gbest model.

There may be different concepts for neighborhood, it can be seen as spatial neighborhood where it is determined by the Euclidean distance between the positions of two particles, or as a sociometric neighborhood (e.g.: the index position in the storing array). They both have social background [10, 11]. The latter is the most commonly used for two main motives: if space coordinates were to represent mental abilities or skills, two very similar individuals may never come to meet in their lifetime, as to elements of the same family, which may differ significantly from each other, but still, they will always be neighbors. The other motive is related with the computational effort required to process the Euclidean distance, when faced with large number of particles or dimensions. At each iteration, the distance between every two particles would have to be calculated and for each particle the nearest k neighbors would have to be sorted out.

IV. ALGORITHM DESCRIPTION FROM LBEST MODEL TO GBEST MODEL

In PSO, particles are directed to global optimum in the problem's solution space. For lbest model and gbest model, their main difference is neighborhood size. The lbest model and gbest model would be attained while we use vary the neighborhood size from 2 to (population size-1). The main pseudo-code for particle-searching is as follows:

Initialize k, c1,c2, and all other parameters;
Initialize individual's position and velocity
Do

$$w = 0.4 + \frac{Iteration\_MaxNum - Iteration\_Num}{2 * Iteration\_MaxNum}$$

Calculate the fitness value of each particle

$\vec{p}_g = \min(\vec{p}_i)$

For i= 1 to Population Size

If $f(\vec{p}_i) > f(\vec{x}_i)$ then $\vec{p}_i = \vec{x}_i$

Find k nearest neighbors

$\vec{p}_l = \min(\vec{p}_{neighbors})$

For d = 1 to Dimension

Update each dimension value of p_i

Next d

Next i

While ($f(\vec{p}_g) > Min\_error$) Or (*Iteration_Num* < *Iterations_MaxNum*)

There are two major measures of performance in an optimizer such as the particle swarm [8]. The first is best function result attained after some number of iterations, such as 600 iterations. It is possible however for the algorithm to rapidly attain a relatively good result while becoming trapped on a local optimum. Thus a second dependent measure was the number of iterations required for the algorithm to meet a criterion. If it doesn't meet by a large iteration time, the measure was considered infinite, that is, it was reported as if the criterion would never be met. So, a third dependent measure was derived from the second. That is a simple binary variable describing whether the version attains the criterion or not. The success rate would be obtained from the average and median iterations for successful runs.

V. BENCHMARK TEST FUNCTIONS

Several types of particle swarms were used to optimize a set of unconstrained real-valued benchmark function. For each of these functions, the goal is to find the global minimiser. Stated formally:

Given f: $\mathrm{R}^n \rightarrow \mathrm{R}$

find $\mathrm{x}^* \in \mathrm{R}^n$ for which f(**x***)≤f(**x**), $\forall \mathrm{x}^* = \mathrm{R}^n$

A. Rosenbrock variant (De Jong's f2) [12]

$$f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

where $-2.048 < x_i < 2.048$ (4)

Continuous, Unimodal; **x***=(1,1), with f(**x***)=0.

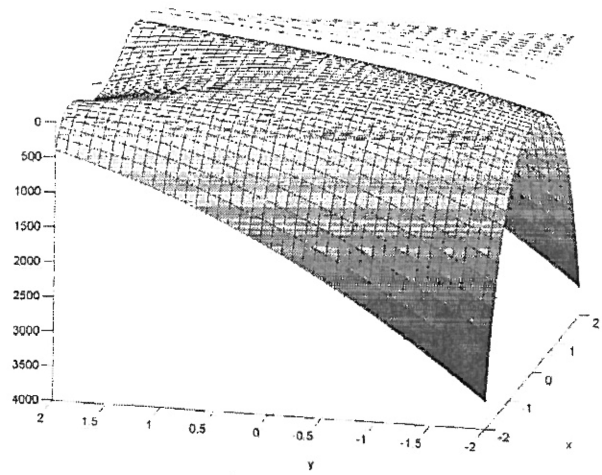

Fig. 1 De Jong's f2

B. Griewank's function [13]

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n}(x_i)^2 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) + 1$$

where $-300<x_i<300$ (6)

Continuous, Multimodal; $\mathbf{x}^*=0$, with $f(\mathbf{x}^*)=0$.

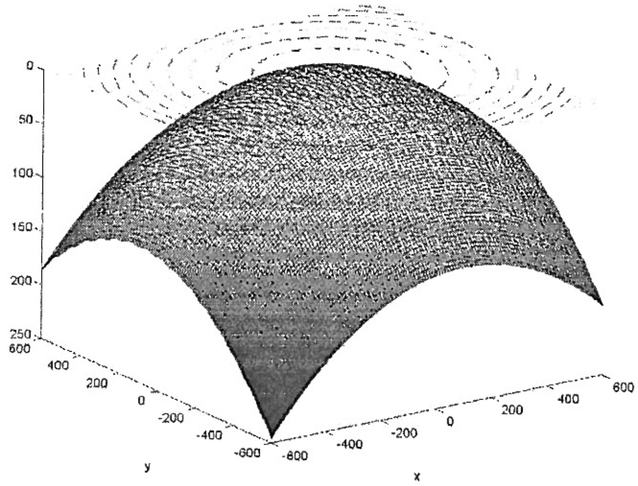

Fig. 2 Globle view of Griewank's function

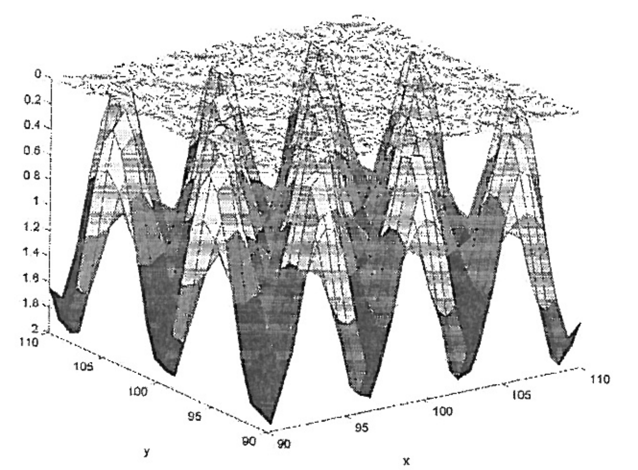

Fig. 3 local view of Griewank's function

VI. Experiments

The performance of PSO is correlative directly with the parameters selection. For getting a good performance in our experiments, all but the neighborhood size are used following the recommended setting in [14, 15]. The inertia weight is implemented as a linearly decreasing function, initially set to 0.9, decreasing to 0.4 over the first 1500 iterations if the iterations are above 1500, and remaining at 0.4 over the remainder of the run. In order to analyze the influence of lbest model to gbest model, we set a smaller value for c1 and a larger value for c2 in our most of experiments. Incorporating Clerc's constriction factor, c2, the coefficient of the social component, is set from 1.0 to 2.4, while c1, the coefficient of the recognition component, is set from 1.5 to 2.0. Vmax set equal to Xmax, and in every experiment, each function was run for 10 repetitions. The success rate was also taken into account. The population size is set to 21.

VII. Results and Discussions

In Fig. 6, 7, the results of iteration procedures display for the two benchmark functions when the neighborhood size is varying. Most of iteration procedures were at standstill state; all particles had got together at one position. But some of iteration procedures still went on, many particles were at different positions dispersedly. The runs were ended because the iteration number partial criterion was met. Since the mean fitness values were used in many other works [8], they were listed in the "mean" column. But in our work they misleaded us in our earlier analysis phase, because the values in lbest model are always larger than that in gbest model. When the fitness values between different groups were compared separately, we could draw a rational conclusion: In general, the gbest model converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while lbest model converges slowly but is able to "flow around" local optima. No surprisingly, the good fitness value appears to depend on the form of the objective function.

We use the sketch map to illuminate the deference between lbest model and gbest, Fig. 4 for unimodal objective functions and Fig. 5 for multimodal objective functions, respectively. In Fig. 4, the fitness value in lbest model is larger than that in gbest model at T1 and T3. If the iteration number partial criterion in the PSO algorithm is met in the run, we would get a larger fitness value in lbest model than that in gbest model. In Fig. 4, iteration number in lbest model is more than that in gbest model at Goalf1 and Goalf2. So the minimum error partial criterion in the PSO algorithm is met in the run, we need use more iterations in lbest model than that in gbest model.

As Fig. 5 indicates the state in lbest model and the gbest model for multimodal objective functions, it is possible for both lbest model and gbest model to arrive local optima due to premature convergences. The gbest model converges quickly but there is a larger probability to trap in local optima, while lbest model spends so much time to explore different regions that it converges very slowly but there is a larger probability to arrival the global optima. In Fig. 5, the fitness value in lbest model is larger than that in gbest model at I1, which is a small iteration time. When we set a small iteration time as the criterion in the PSO algorithm in order to that the iteration time partial criterion is met in the run, we get a larger fitness value in lbest model than that in gbest model. But the results would be reversed possibly at I2 and I3, which are more iteration numbers. In Fig. 5, iteration number in lbest model is more than that in gbest model at Goalf1 and Goalf2 too. When we set larger goal fitness values as the criterion in the PSO algorithm in order to that the iteration number partial criterion is met in the run, we need a more iteration number in lbest model than that in gbest model. But if we set larger goal fitness values as the criterion in the PSO algorithm and there is no other partial criterion, the success rates would be much more than that in gbest model possibly.

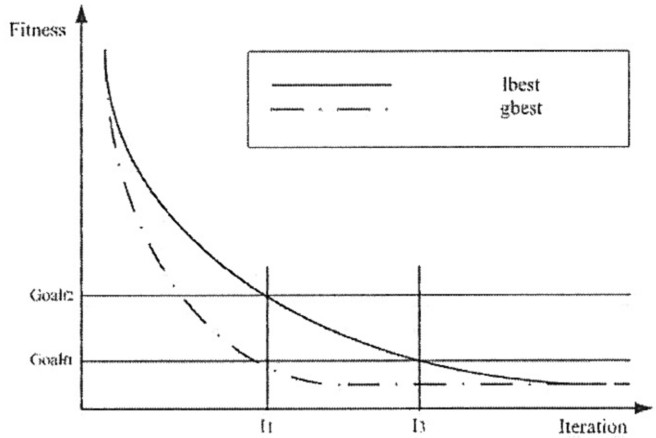

Fig. 4 PSO: from lbest to gbest model for unimodal function

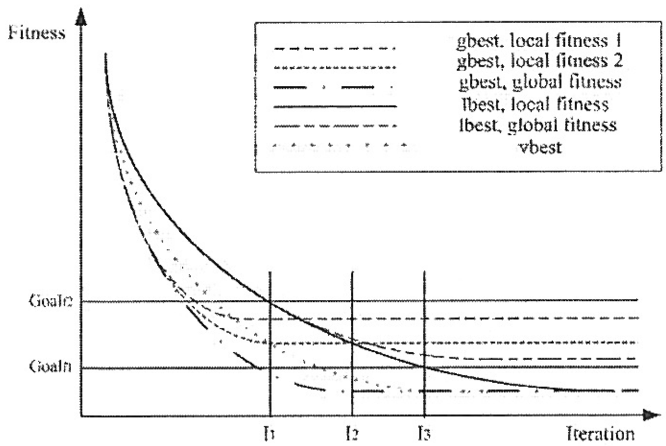

Fig. 5 PSO: from lbest to gbest model for multimodal function

For unimodal objective functions, gbest model is recommended strongly, because the best fitness values of gbest model are almost similar to that of lbest model, and gbest model converges much quickly. For multimodal objective functions, the user can take well-informed decisions according to the desired exploration–exploitation trade-off: either favour exploration by a thorough sampling of the solution space for a robust location of the global optimum at the expense of a large number of objective function evaluations or, on the contrary, favour exploitation resulting in a quick convergence but to a possibly non-optimal solution. If PSO would be used to attain better fitness values for multimodal functions, we recommend a variable neighborhood model, "vbest model", to inherit both quick convergence in gbest model and "flow around" local optima in lbest mode. At the beginning procedure, the number of neighbors, k, is 2, and increasing gradually to (swarm_population_size-1) while iteration number is increasing. The fitness-iteration procedure of vbest model is similar to the omit curve (vbest) in Fig. 5.

VIII. Conclusion

The effects of various neighborhood models on the particle swarm algorithm had been systematically investigated in this paper. We also gave some additional insight into the PSO neighbourhood model selection topic. Our experiment results testified that gbest model converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while lbest model converges slowly on problem solutions but is able to "flow around" local optima, as the individuals explore different regions. The gbest model is recommended strongly for unimodal objective functions, while a variable neighborhood model is recommended for multimodal objective functions.

Acknowledgment

The first author wishes to thank to Bo Li, Ye Ji, and Ran He, who listened patiently when I discussed some of my ideas with them, for their feedback and insight.

References

[1] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on neural networks, Perth, Australia, pp.1942-1948, 1995
[2] R. C. Eberhart, and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", Proceedings of 7th annual conference on evolutionary computation, pp.611-616, 1998
[3] J. Kennedy, R. Eberhart, Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001
[4] P.N. Sugarthan, "Particle Swarm Optimiser with Neighbourhood Operator", 1999 Congress on Evolutionary Computing, 1999, Vol.III, pp.1958-1964,
[5] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance", Proceedings of IEEE conference on Evolutionary Computation, 1999, pp.1931-1938,
[6] A.Carlisle and G. Dozier, "An off-the-shelf PSO", Proceedings of the Particle Swarm Optimization Workshop, Indianapolis, IN, 2001
[7] E. S. Peer, F. van den Bergh, and A. P. Engelbrecht, "Using Neighborhoods with the guaranteed Convergence PSO", Proceeding of IEEE conference on Evolutionary Computation, 2003, pp.235-242,
[8] J. Kennedy, and R. Mendes, "Population structure and particle swarm performance", Proceeding of IEEE conference on Evolutionary Computation, 2002, pp.1671-1676
[9] Y. Shi, and R. C. Eberhart, , "A modified particle swarm optimizer", Proceedings of the International Joint Conference on Evolutionary Computation, 1998, pp.69-73
[10] D.J. Watts and S. H. Strogatz, Collective dynamics of 'Small-World' networks, Nature,393, pp.40-442, 1998
[11] D.J. Watts, Small World: The dynamics of networks between order and randomness, Princeton Unibersity Press, 1999

[12] F. Seredynski, P. Bouvry and F. Arbab, "Distributed Evolutionary Optimization in Manifold: the Rosenbrock's Function Case Study", Proceeding of First International Workshop on Frontiers in Evolutionary Algorithms, Duke University, USA, 1997

[13] M. Clerc, and J. Kennedy, "The particle swarm—explosion, stability, and convergence in a multidimensional complex space", IEEE Transactions on Evolutionary Computation, 2002, V6(1), pp.58-73

[14] R. C. Eberhart, and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", Proceedings of IEEE International Congress on Evolutionary Computation, 2000, pp.84-88

[15] I.C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection", Information Processing Letters, 2003, 85, pp.317-325

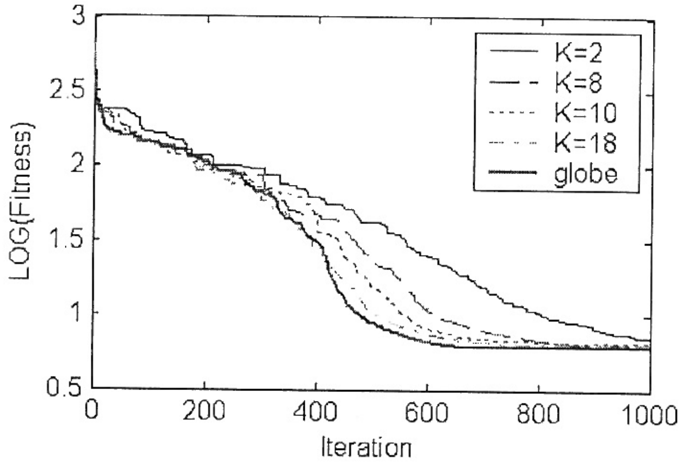

Fig. 6 Iteration procedures in PSO for De Jong's f2

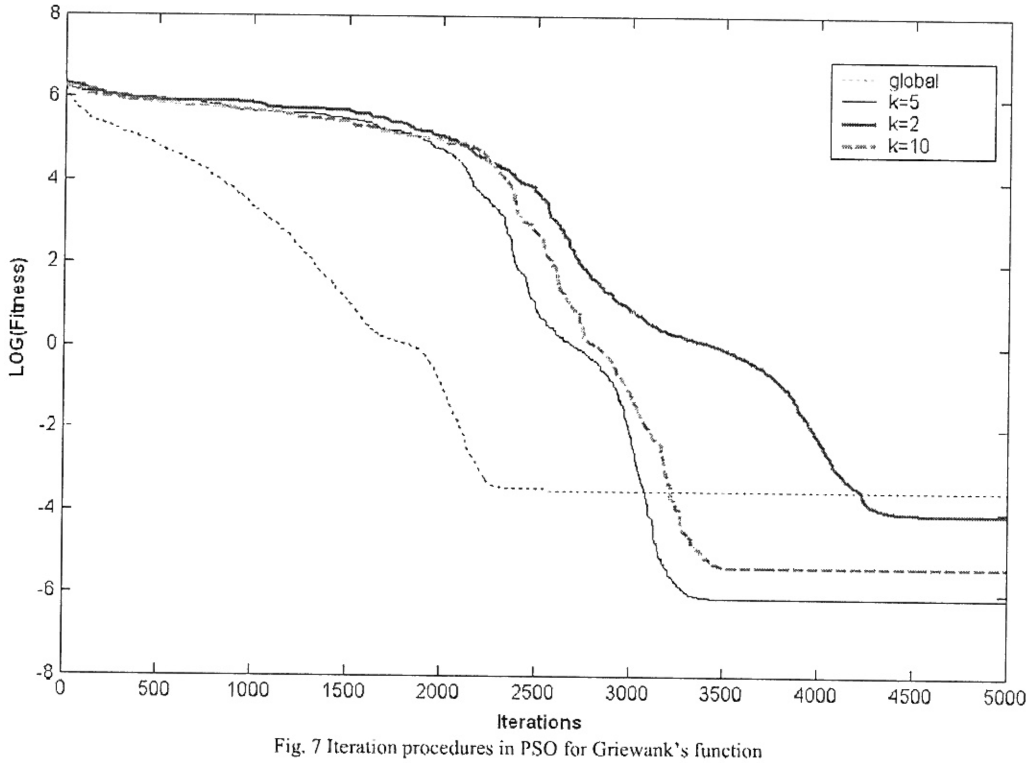

Fig. 7 Iteration procedures in PSO for Griewank's function