Int. J. Advance. Soft Comput. Appl., Vol. 2, No. 1, March 2010
ISSN 2074-8523; Copyright © ICSRS Publication, 2010
www.i-csrs.org

Combined Heuristic Optimization Techniques for Global Minimization

Premalatha K, Natarajan A M

Bannari Amman Institute of Technology, Erode, Tamil Nadu, India
e-mail: kpl_barath@yahoo.co.in

Bannari Amman Institute of Technology, Erode
Tamil Nadu, India

Abstract

This paper presents Combined Heuristic Optimization Techniques of Particle Swarm Optimization (PSO) algorithm with Simulated Annealing (SA). Particle Swarm Optimization is Swarm Intelligence based algorithm to find a solution to an optimization problem in search space. SA is a generic probabilistic metaheuristic for locating the global minimum of a given function in a large search space. In standard PSO the non-oscillatory route can quickly cause a particle to stagnate and also it may prematurely converge on suboptimal solutions that are not even guaranteed to local optimal solution. The proposed system improves the solution by incorporating the working principles of SA to Standard PSO to diversify the particle position. Experiment results are examined with benchmark functions. It demonstrates that the proposed PSO outperforms the standard PSO

Keywords: *Convergence, Global Minimum, PSO, Simulated Annealing, Stagnation*

1 Introduction

The basic optimization problem is that of minimizing or maximizing an objective function subject to constraints imposed on the variables of that function. The objective function and constraints can be linear or nonlinear; the constraints can be bound constraints, equality or inequality constraints, or integer constraints.

Global optimization is the task of finding the absolutely best set of admissible conditions to achieve an objective under given constraints. Global optimization is just a stronger version of local optimization, whose great usefulness in practice is definite. Instead of searching for a locally unemployable feasible point one wants the globally best point in the feasible region.

The global minimization problem can be defined as follows (Paradalos et al 2002). Given $f: S \rightarrow R$ where $S \subseteq \Re^{N_d}$ and N_d is the dimension of the search space *S*. Find $y \in S$ such that $f(y) \leq f(z), \forall z \in S$. The variable *y* is called the global minimizer of *f* and *f(y)* is called the global minimum value of *f*. The process of finding the global optimal solution is known as global optimization (Gray et al. 1997). A true global optimization algorithm will find *y* regardless of the selected starting point $z_0 \in S$ (Van den Bergh 2002). The variable y_C is called the local minimizer of *C* because $f(y_C)$ is the smallest value within a local neighborhood, *C*. Mathematically speaking the variable y_C is a local minimizer of the region *C* if $f(y_C) \leq f(z), \forall z \in C$ where $C \subset S$.

Every global minimizer is a local minimizer, but a local minimizer is not necessarily a global minimizer y_C of the region *C* if a starting point z_0 is used with $z_0 \in C$. An optimization algorithm that converges to a local minimizer, regardless of the selected starting point $z_0 \in S$, is called a global convergent algorithm. Generally, a local optimization method is guaranteed to find the local minimizer. In this study, finding global minimum solution using PSO and SA is proposed. Section 2 describes an overview of SA approach. Standard PSO is discussed in Section 3. Section 4 gives the hybrid of PSO and SA. Section 5 presents the detailed experimental setup and results for comparing the performance of the proposed algorithm with the simple PSO.

2 Simulated Annealing

In an optimization problem, often the solution space has many local minima. A simple local search algorithm proceeds by choosing random initial solution a d generating a neighbor from that solution. If it is a minimum fitness transition then the neighboring solution is accepted. Such an algorithm has the drawback of often converging to a local minimum. The simulated annealing algorithm avoids getting trapped in a local minimum by accepting cost increasing neighbors with some probability. In SA, first an initial solution is randomly generated, and a neighbor is found and is accepted with a probability of min (1, exp (-ΔE/T)), where ΔE is the cost difference and T is the control parameter corresponding to the temperature of the physical analogy and will be called temperature On slow reduction of temperature, the algorithm converges to the global minimum. Among its advantages are the relative ease of implementation and the ability to provide reasonably good solutions for many combinatorial problems. Simulated Annealing is inherently sequential and hence very slow for problems with large search spaces. Though a robust technique, its drawbacks include the need for a great deal of computer time for many runs and carefully chosen tunable parameters.

Algorithm

```
Set X a initial configuration
Set E as Eval(X)
Set T as high temperature and frozen is false
while (!frozen)
        repeat
                Choose a random move i from the move set
                Set Ei as Eval(move(X, i))
                if E < Ei then
                        set X as move(X, i)
                        set E as Ei
                else accept the move  with probability exp(-(ΔE/T)  even though
                things get worse
        until the s stem is in thermal equilibrium at T
        if ((E is still  decreasing over the last few temperatures)
                reduce T
        else
                assign frozen is true
```

3 Particle Swarm Optimization

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. Particle Swarm Optimization (PSO) incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged (Kennedy, 2001) (Clerc , 2002), (Parsopoulos, 2004). PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems. As an algorithm, the main strength of PSO is its fast convergence, which compares favorably with many global optimization algorithms like Genetic Algorithms (GA) (Goldberg, 1989) Simulated Annealing (SA) (Orosz, 2002), (Triki, 2005) and other global optimization algorithms. For applying PSO successfully, one of the key issues is finding how to map the problem solution into the PSO particle, which directly affects its feasibility and performance.

The original PSO formulae define each particle as potential solution to a problem in *D*-dimensional space. The position of particle *i* is represented as

$$X_i = (x_{i1}, x_{i2}, \ldots\ldots\ldots\ldots, x_{iD})$$

Each particle also maintains a memory of its previous best position, represented as

$$P_i = (p_{i1}, p_{i2}, \ldots\ldots\ldots\ldots, p_{iD})$$

A particle in a swarm is moving; hence, it has a velocity, which can be represented as

$$V_i = (v_{i1}, v_{i2}, \ldots\ldots\ldots\ldots, v_{iD})$$

Each particle knows its best value so far (***pbest***) and its position. Moreover, each particle knows the best value so far in the group (***gbest***) among ***pbest***s. This information is analogy of knowledge of how the other particles around them have performed. Each particle tries to modify its position using the following information:

- the distance between the current position and pbest
- the distance between the current position and gbest

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation (1) in inertia weight approach (IWA)

$$v_{id} = w * v_{id} + c_1 * r1 * (P_{id} - X_{id}) + c_2 * r2 * (P_{gd} - X_{id}) \quad (1)$$

where,

v_{id} : velocity of particle
x_{id} : current position of particle
w : inertia factor,
c_1 & c_2 : determine the relative influence of the social and cognitive components
p_{id} : pbest of particle i,
p_{gd} : gbest of the group
$r1, r2$: random numbers

Where w is called as the inertia factor which controls the influence of previous velocity on the new velocity, r_1 and r_2 are the random numbers, which are used to maintain the c_1 is a positive constant, called as coefficient of the self-recognition component; c_2 is a positive constant, called as coefficient of the social component. From equation (1), 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 particle in the swarm. In the particle swarm model, the particle searches the solutions in the problem space with a range [$-s$, s]

The following inertia factor is usually utilized is shown in equation (2)

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} x\, iter \tag{2}$$

where,

w_{max} : initial weight,
w_{min} : final weight,
$iter_{max}$: maximum iteration number,
$iter$: current iteration number.

Using the above equation, diversification characteristic is gradually decreased and a certain velocity, which gradually moves the current searching point close to *pbest* and *gbest* can be calculated. The current position (searching point in the solution space) can be modified by means of the equation (3):

$$X_{id} = X_{id} + V_{id} \tag{3}$$

All swarm particles tend to move towards better positions; hence, the best position (i.e. optimum solution) can eventually be obtained through the combined effort of the whole population.

Maurice Clerc has introduced a constriction factor *k,* (CFA) that improves PSO's ability to constrain and control velocities. *k* is computed as:

$$k = \frac{2}{|\, 2 - \mathrm{c} - \sqrt{c^2 - 4c}\, |} \tag{4}$$

Where

$c = c_1 + c_2$ and c > 4

$$V_{id} = k(V_{id} + c_1 * rand() * (P_{id} - X_{id}) + c_2 * rand() * (P_{gd} - X_{id})) \tag{5}$$

For example, if c=4.1, then k=0.729. As c increases above 4.0, k gets smaller. For example, if c=5.0, then k=0.38, and the damping effect is even more pronounced.

Algorithm

1. Initialize the population - positions and velocities
2. Evaluate the fitness of the individual particle (*pbest*)
3. Keep track of the individuals highest fitness (*gbest*)
4. Modify velocities based on *pbest* and *gbest* position
5. Update he particles position
6. Terminate if the condition is met
7. Go to Step 2

Figure 1 shows Standard PSO.

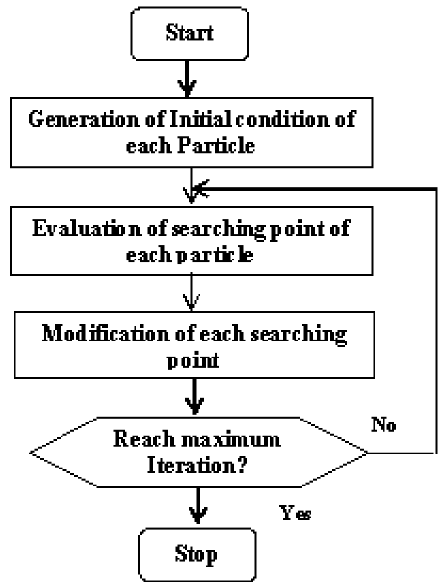

Fig. 1. Standard PSO

4 Literature Review of Hybrid PSO

Holden et al (2007) proposed hybrid PSO/ACO algorithm for classification. Unlike a conventional PSO algorithm, this hybrid algorithm can directly cope with nominal attributes, without converting nominal values into numbers in a pre-processing phase. The design of this hybrid algorithm was motivated by the fact that nominal attributes are common in data mining.

De-Shuang Huang et al (2007) proposed hybrid PSO algorithm with the Sequential Quadratic Programming (SQP). In the PSO-SQP algorithm, PSO algorithm is the basic optimizer and the SQP technique is used to reduce computation time and improved convergence performance.

Cui and Potok (2005) proposed a PSO based hybrid document clustering algorithm. The PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments, they applied the PSO, K-Means and a hybrid PSO+K-Means clustering algorithm on four different text document datasets. The results illustrated that the hybrid PSO algorithm can generate more compact clustering results than the K-Means algorithm. Geetha and Michael (2009) proposed hybrid PSO and K-Means algorithm for gene expression data clustering.

Tian et al (2009) proposed a hybrid method by combining two heuristic optimisation techniques, PSO and ElectroMagnetism-like (EM) mechanism, called PSO-EM, for the global optimisation of functions. This hybrid technique incorporates concepts of PSO and EM and creates individuals in a new generation not only by features of PSO, but also by attraction-repulsion mechanism of EM. Premalatha and Natarajan (2009) presented a hybrid PSO with GA for global maximization. When the PSO global best particle stagnates, the GA operators are applied to change the position of the particle.

Liu and Qui (2009) proposed the hybrid PSO-BP algorithm which combines the PSO mechanism with the Levenberg-Marquardt algorithm or the Conjugate gradient algorithm. The main idea employed, BP algorithm with numeric technology to find the local optimum, and taken the weights and biases trained as particles, and harnesses swarm motion to search the optimum. Finally, the hybrid algorithm selected some good particles from the local optimum set to predict the new samples. Chung and Lau (2009) presented a hybrid optimization based on the PSO and Differential Evolution (DE) algorithms to manage the Dynamic Economic Dispatch (DED). The hybrid approach incorporated DE operators into the PSO model to enrich the information exchanges amongst candidate solutions.

Zhao et al (2005) proposed Hybrid PSO with SA for partner selection in virtual enterprises. In this system, the PSO operations are performed first with the maximum number of iterations. Afterwards, the SA is applied on *gbest* particle of the swarm. The proposed system combines SA, when the particle stagnates in the local optimal solution. It causes the diversity in swarm.

5 The Proposed SAPSO methods

The proposed system combines PSO and SA for global minimization. The drawback of PSO is that the swarm may prematurely converge. The underlying principle behind this problem is that, for the global best PSO, particles converge to a single point which is on the line between the global best and the personal best positions. This point is not guaranteed to be even a local optimum (Van den Bergh 2002). Another reason for this problem is the fast rate of information flow between particles, resulting in the creation of similar particles (with a loss in diversity) which increase the possibility of being trapped in local optima.

A further drawback is that stochastic approaches have problem-dependent performance. This dependency usually results from the parameter settings in each algorithm. The different parameter settings for a stochastic search algorithm result in high performance variances. In general, no single parameter setting exists which can be applied to all problems. Changing the parameter value will increase the speed of the particles resulting in more exploration (global search) and less exploitation (local search) or on the other hand, more exploitation and less exploration. Thus finding the best value for the parameter is not an easy task and it may differ from one problem to another. Therefore, from the above it can be concluded that the PSO performance is problem-dependent. The problem-dependent performance can be addressed through hybrid mechanism. Hybrid refers to combining different approaches to benefit from the advantages of each approach.

Simulated Annealing (SA) is locating a good approximation to the global minimum of a given function in a large search space. Each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter *T* called the temperature that is gradually decreased during the process.

The proposed system incorporates the SA in PSO with three different strategies

a. Every iteration of PSO
b. Only when the individual particles stagnate in their *pbest* position over a period of time, the number of iterations taken for stagnation checking is 5
c. Only when the *gbest* particle stagnates, the number of iterations taken for stagnation checking is 5

Algorithm for SAPSO when gbest particle stagnates

1. Initialize temperature
2. Initialize the population - positions and velocities
3. Evaluate the fitness of the individual particle (*pbest*)
4. Keep track of the individuals highest fitness (*gbest*)
5. Modify velocities based on *pbest* and *gbest* position
6. Update the particles position
7. If *gbest* position is not changed over a period of time
    a. Find a new position using temperature
    b. accept the new position as *gbest* position with probability exp(-(ΔE/T) even though current position is worse
8. reduce T
9. Terminate if the condition is met
10. Go to Step 3

Figure 2 shows *SAPSO* when gbest particle stagnates

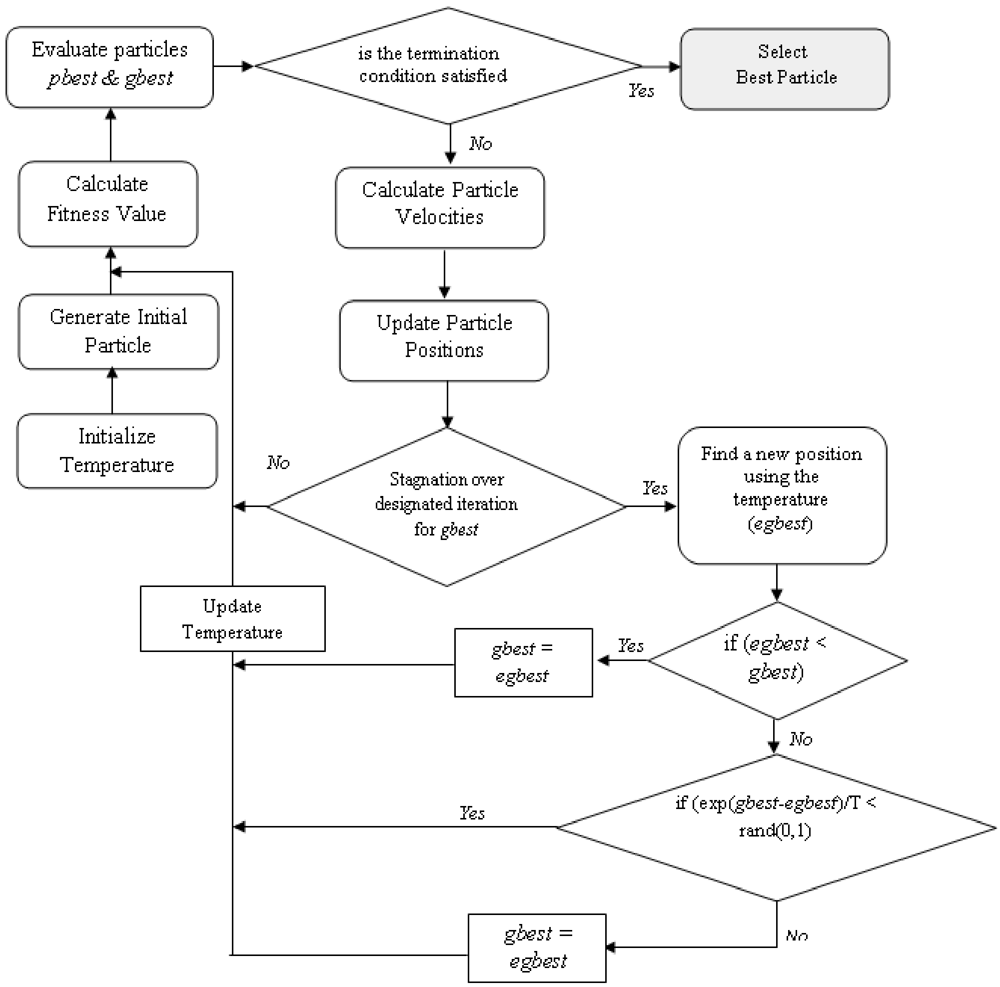

Fig. 2. SAPSO when *gbest* particle stagnates

5. Experiment results

For comparison, five benchmark functions are taken from evolutionary computation literature (Yao et al, 1999) which is shown in Table 1. Except De Jong's all functions are high-dimensional problems. Functions Schwefel's and Rosenbrock are unimodal. Rastrigin and Griwank are multimodal functions where the number of local minimum increases exponentially

For the proposed system, c_1 and c_2 are assigned as 2.1 and w is 0.9. The initial population is generated from a uniform distribution in the range specified. The results reported for 100 iterations and the number of particles that used is 10. Temperature T is set based on the function. Figures 3-7 show the result obtained from Simple PSO, SAPSO All (Applying SA in all iterations), SAPSO in Pbest (Applying SA, only when individual particle stagnates) and SAPSO (Applying SA, only when *gbest* particle stagnates). For all the given five functions which are shown in Table 1, fitness value obtained from SAPSO as better fitness value than Simple PSO, SAPSO in Pbest and SAPSO All. It shows that the proposed hybrid mechanism of PSO with SA gives better performance only when the *gbest* particle stagnates. However, the remaining two methods unnecessarily change the particle position; they give better result than Simple PSO. Table 2 shows the comparison of experiment results obtained from simple PSO and the proposed system. The SAPSO Count 5 gives the minimum fitness value for all the given functions.

Table 1: Benchmark functions

Name	Functions	Dimension	Initial range of x_i
De Jong's	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	2	±50
Rastrigin	$f(x) = \sum_{d=1}^{D} \left(x_d^2 - 10\cos(2\prod x_d) + 10\right)$	Low = 10 High = 30	±5.12
Griewank	$f(x) = \frac{1}{4000}\sum_{d=1}^{D} x_d^2 - \prod_{d=1}^{D} \cos\left(\frac{x_d}{d}\right) + 1$	Low = 10 High = 30	±300
Rosenbrock	$f(x) = \sum_{d=1}^{D} (100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2$	Low = 10	±10
Schwefel's	$f(x) = \sum_{d=1}^{D} \lvert x_d \rvert + \prod_{d=1}^{D} \lvert x_d \rvert$	10	±10

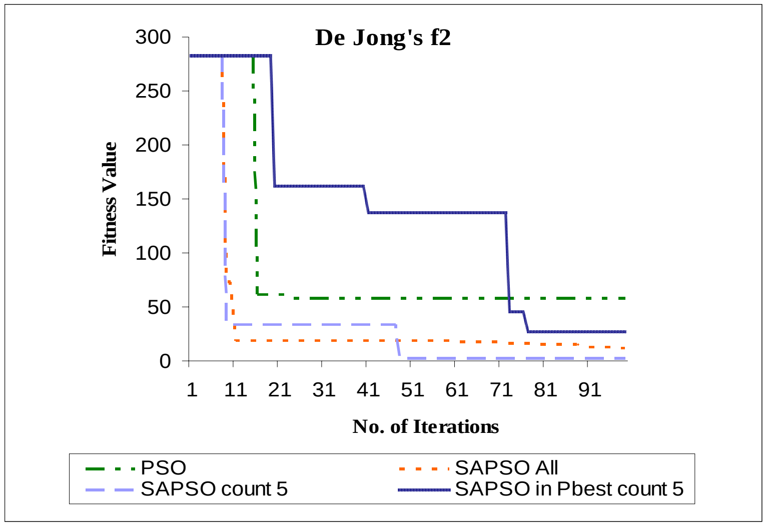

Fig. 3. Fitness value obtained from De Jong's function

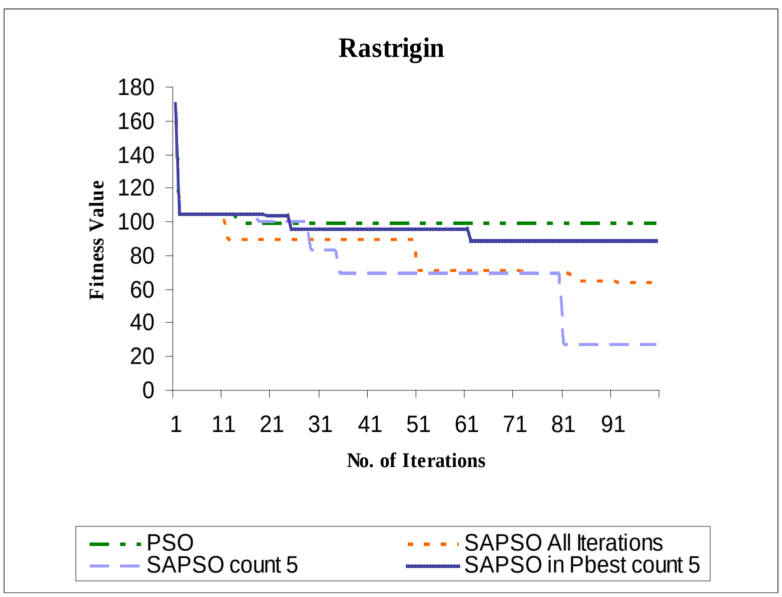

Fig. 4. Fitness value obtained from Rastrigin function

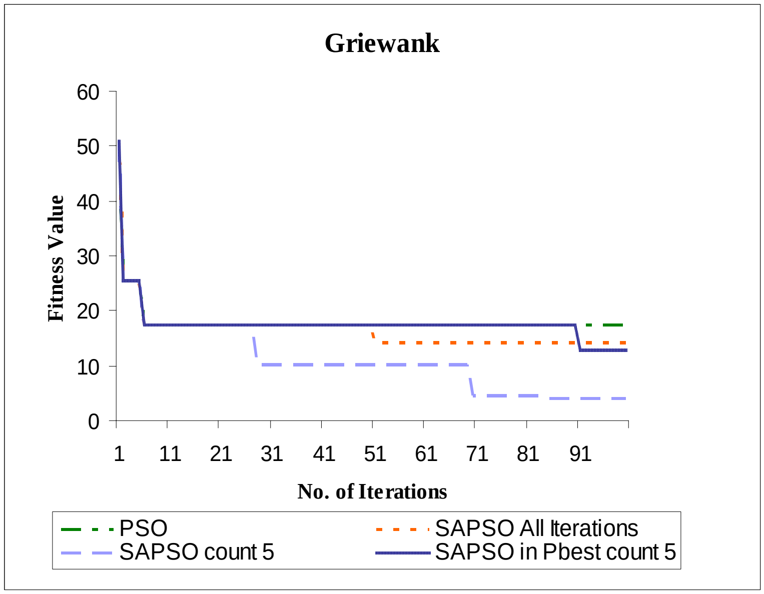

Fig. 5. Fitness value obtained from Griewank function

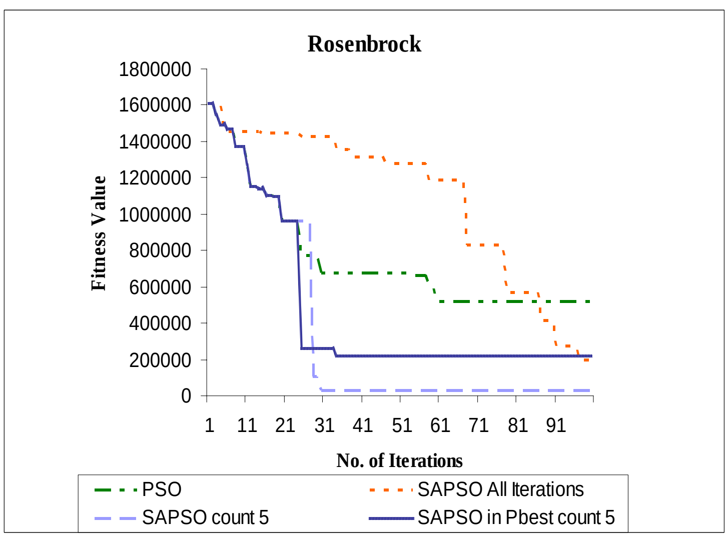

Fig. 6. Fitness value obtained from Rosenbrock function

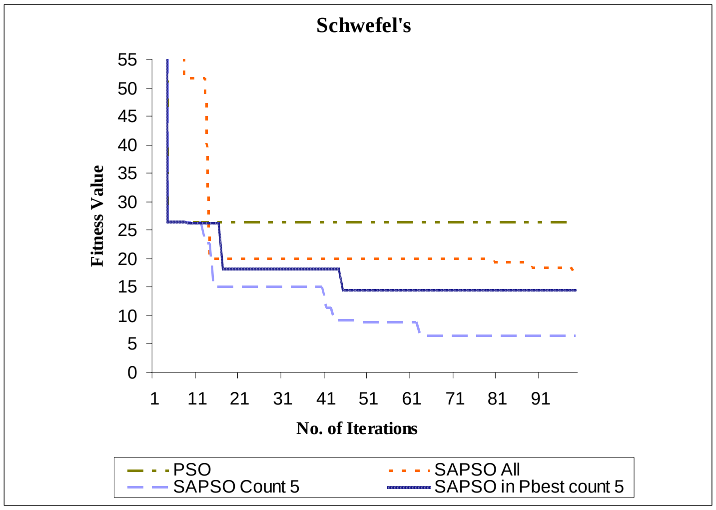

Fig.7. Fitness value obtained from Schwefel function

Table 2: Comparison results of PSO and proposed system

Name	Fitness Value				% of Improvement		
	PSO	SAPSO in all iterations	SAPSO Count 5	SAPSO in Pbest Count 5	SAPSO in all iterations (%)	SAPSO Count 5 (%)	SAPSO in Pbest Count 5 (%)
De Jong's	57.8010	11.3462	**2.4801**	26.8808	80.37	95.71	53.49
Rastrigin	98.9708	59.8801	**27.4149**	88.6184	39.50	72.30	10.46
Griewank	17.4099	14.1913	**3.9074**	12.9471	18.49	77.56	25.63
Rosenbrock	520155.29	197757.67	**26402.44**	215300.90	61.98	94.92	58.61
Schwefel's	26.4797	17.1777	**6.4338**	14.4416	35.13	75.70	45.46

6. Open Problems

There is no improvement on the *gbest* position of the swarm over several time steps is called stagnation. The result obtained from PSO during stagnation is not even local optimal solution. Also PSO needs some parameter settings; the user desires to define some parameters. Depends on the parameter settings the fitness value achieved and the number of fitness evaluations are changed. Optimizing the Optimizer is essential.

7. Conclusion

PSO, which is stochastic in nature and makes use of the memory of each particles as well as the knowledge gained by the swarm as a whole, has been proved to be powerful in solving many optimization problems. The proposed hybrid PSO algorithm finds a better solution without trapping in local minimum, and to achieve faster convergence rate. This is because when the PSO particles stagnate, SA concept diversifies the particle position even though the solution is worse. In SAPSO, particle movement uses randomness in its search. Hence, it is a kind of stochastic optimization algorithm that can search a complicated and uncertain area. This makes SAPSO more flexible and robust than conventional methods. Unlike standard PSO, SAPSO is more reliable in giving better quality solutions with reasonable computational time, since the hybrid strategy avoids premature convergence of the search process to local optima and provides better exploration of the search process.

References

[1] Chung, T.S. Lau, T.W., "Hybrid PSO and DE Approach For Dynamic Economic Dispatch with Non-Smooth Cost Functions", *International Journal of Modelling, Identification and Control*, Vol 8, No. 4, (2009) , pp. 317-326(

[2] Cui X. and Potok T.E. "Document Clustering Analysis based on Hybrid PSO+K-means Algorithm", *Journal of Computer*, (2005), pp. 27-33.

[3] De-Shuang Huang, Laurent Heutte and Marco Loog, *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, Springer Berlin Heidelberg, (2007), pp. 596-604

[4] Gray P., Hart W.E., Painton L., Phillips C., Trahan M. and Wagner J., A Survey of Global Optimization Methods, Technical Report, Sandia National Laboratory, (1997)

[5] Geetha T and Michael A, "Effective Hybrid PSO and K-Means Clustering Algorithm for Gene Expression Data", *International Journal of Rapid Manufacturing* Vol. 1, No.2, (2009) pp. 173 - 188

[6] Holden, Nicholas Paul and Freitas, Alex A., “A Hybrid PSO/ACO Algorithm for Classification” *Proceedings of the 2007 GECCO conference companion on Genetic and Evolutionary Computation*, (2007) pp. 2745-2750

[7] Jun Liu, Xiaohong Qiu, "A Novel Hybrid PSO-BP Algorithm for Neural Network Training," *International Joint Conference on Computational Sciences and Optimization* , vol. 1, (2009), pp.300-303

[8] Van den Bergh F. and Engelbrecht A.P., “A Cooperative Approach to Particle Swarm Optimization”, *IEEE Transactions on Evolutionary Computation*, (2004), pp. 225-239.

[9] Kennedy J and Eberhart R, *Swarm intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, CA,(2001)

[10] Clerc M and Kennedy J, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space”. *IEEE Transactions on Evolutionary Computation*, Vol 6, No 1, pp. 58-73

[11] Goldberg D E, *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Corporation, Inc, (1989)

[12] Tian, Ye, Liu, Dayou, Ma, Xin, Zhang, Changsheng “A Hybrid PSO with EM for Global Optimization”. *International Journal of Modelling, Identification and Control*, Volume 8, Number 4, (2009) , pp. 327-334

[13] Triki E, Collette Y and Siarry P, “A theoretical study on the behavior of simulated annealing leading to a new cooling schedule”. *European Journal of Operational Research*, Vol. 166, (2005), pp. 77-92.

[14] Ordonez C, “Clustering Binary Data Streams with K-means”, *In Proceedings of DMKD*, (2003), pp. 12-19.

[15] Premalatha K and Natarajan A.M., “Hybrid PSO and GA for Global Maximization” *International Journal Open Problems Compt. Math.*, Vol. 2, No. 4, 2009,pp. 597 – 608

[16] Yao, X., Liu, Y., and Lin, G. “Evolutionary programming made faster”. *IEEE Transactions on Evolutionary Computation*, , (1999), 3(2): pp 82–102

[17] Zhao, F. Q., Zhang Q. Y., Yu, D. M., Chen, X. H. Yang, Y. H. “A hybrid algorithm based on PSO and simulated annealing and its applications for partner selection in virtual enterprises”. Advances in Intelligent Computing, PT 1, *Proceedings Lecture Notes in Computer Science.* 3644, (2005), pp. 380–38