# Adaptive Particle Swarm Optimization (APSO) for multimodal function optimization

Md. Sakhawat Hossen [1], Fazle Rabbi [2] Md. Mainur Rahman [3]

[1] *The Royal Institute of Technology (KTH)*
*Stockholm, Sweden*
[1]hossen@kth.se

[2] *St. Francis Xavier University*
*54B St. Mary's Street. Canada*
[2]rabbi.uign@gmail.com

[3]*University of Calgary*
*NW Calgary, Alberta, Canada*
[3]mmrahma@ucalgary.ca

***Abstract*— This research paper presents a new evolutionary optimization model based on the particle swarm optimization (PSO) algorithm that incorporates the flocking behavior of a spider. The search space is divided into several segments like the net of a spider. The social information sharing among the swarms are made strong and adaptive. The main focus is on the fitness of the swarms adjusting to the learning factors of the PSO. The traditional Particle Swarm Optimization algorithms converges rapidly during the initial stage of a search, but in course of time becomes steady considerably and can get trapped in a local optima. On the other hand in the proposed model the swarms are provided with the intelligence of a spider which enables them to avoid premature convergence and also help them to escape from local optima. The proposed approaches have been validated using a series of benchmark test functions with high dimensions. Comparative analysis with the traditional PSO algorithm suggests that the new algorithm significantly improves the performance when dealing with multimodal functions.**

***Keywords: Particle swarm optimization (PSO), evolutionary algorithm, multimodal function.***

## I. INTRODUCTION

Inspired by the natural flocking as well as swarm behavior of birds and insects, the concept of Particle Swarm Optimization (PSO) was emerged and has gained its popularity since 1995[1]. The PSO algorithm was first introduced by Eberhart and Kennedy[2][3] that has proved to be a very useful algorithm to optimize unconstrained functions. It is an evolutionary computation model which is based on swarm intelligence. Since 1995 many attempts have been made to improve the performance of the original PSO. Particle swarm optimization shares many features with Genetic Algorithms and Evolutionary Paradims [4].

Population based stochastic algorithms does require only the function values. It requires neither the function derivatives nor any other extra information about the problem. For this reason, stochastic search algorithms have been a very popular choice in the recent years in the research arena of computational intelligence [7]. Some well established search algorithms such as Genetic Algorithm (GA) [8] [9] [10], Evolutionary Strategies (ES) [11], Evolutionary Programming (EP) [12] and Artificial Immune Systems (AIS) [13], have been successfully implemented to solve optimization problems.

PSO is a population-based heuristic search technique in which each particle represents a solution within the search space. It is a population-based stochastic optimization method. Each particle has a position in the search space, a velocity and a record of its past performance. The initial population is distributed randomly over the search space. In course of flight, every individual particle searches for the optimal value of a function by updating its position through a number of generations. Every particle follows a simple equation to update its position and velocity. The traditional PSO does not use crossover and mutation operator. Some improved PSO algorithms have been developed. A hybrid PSO algorithm was proposed by adding the idea of Genetic Algorithm (GA) in [5] and [6]. This paper presents a modified algorithm where several modifications have been performed on the traditional PSO algorithm, which improves the performance to solve optimization problems.

The rest of the paper is organized as follows. The traditional Particle Swarm Optimization (PSO) algorithm is described in section II. The proposed algorithm is presented in section III. Section IV provides the evaluation of the proposed algorithm and the comparison analysis with the traditional PSO algorithm and finally Section V concludes the paper.

## II. PARTICLE SWARM OPTIMIZATION(PSO)

PSO originated from the research of food hunting behaviours of birds. Researchers found that in the course of flight flocks of birds would always suddenly change direction, scatter and gather. Through the research of the behaviours of similar biological communities, it is found that there exists a social information sharing mechanism in biological communities. This mechanism provides an advantage for the evolution of biological communities, and provides the basis for the formation of PSO.

Every swarm of PSO is a solution in the solution space. It adjusts its flight according to its own and its companion's flying experience. The best position in the course of flight of each swarm is the best solution that is found by the swarm. The best position of the whole flock is the best solution, which is found by the flock. The former is called *pBest*, and the latter is called *gBest*. Every swarm continuously updates itself through the above mentioned best solution. Thus a new generation of community comes into being. In the practical operation, the fitness function, which is determined by the optimization problem, assesses the extent to which the swarm is *good* or *bad*. Obviously, each swarm of PSO can be considered as a point in the solution space. If the scale of swarm is $N$, then the position of the $i$-th (i=1,2,....N) particle is expressed as $X_i$ . Total swarms particles are a set S = {$x_1$ , $x_2$ ,......., $x_N$ }. The "best" position passed by the particle is expressed as *pBest[i]* . The speed is expressed with $V_i$. The index of the position of the "best" particle of the swarm is expressed with *g*. Therefore, swarm *i* will update its own speed and position according to the following equations:

$$V_i = w*V_i + c_1 * rand() * (pBest[i] - X_i) + c_2 * Rand() * (pBest[g] - X_i) \quad \ldots\ldots\ldots.. (2.1)$$

$$X_i = X_i + V_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots... (2.2)$$

During each iteration of the PSO algorithm, the personal best $y_i$ of each particle is compared to its current performance, and set to their better performance. If the objective function is to be minimized is defined as $f$: $R^n \rightarrow R$, then

$$y_i = \begin{cases} y_i & \text{if } f(x_i) \geq f(y_i) \\ x_i & \text{if } f(x_i) < f(y_i) \end{cases} \quad (2.3)$$

$$\hat{y} \in \{ y_1 , y_2 , \ldots .y_N \mid f(\hat{y}) = \min (f(y_1) , f(y_2) , \ldots\ldots f(y_N) ) \} \quad (2.4)$$

Where $c_1$ and $c_2$ are the learning factors which are two positive constants, *rand ()* and *Rand ()* are two random numbers within the range (0:1), and w is the *inertia weight*. The most common settings for $c_1$ and $c_2$ are $c_1$ = $c_2$ = 2.0. Inertia weight w typically has values in the range (0, 1) with improvements in the convergence properties being observed when the value of w is reduced linearly from 0.9 to 0.4 over the number of generations of the search [14]. The equations consist of three parts. The first part is the former speed of the swarm, which shows the present state of the swarm; the second part is the cognition modal, which expresses the thought of the swarm itself; the third part is the social modal. The three parts together determine the space searching ability. The first part has the ability to balance the whole and search a local part. The second part causes the swarm to have a strong ability to search the whole and avoid local minimum. The third part reflects the information sharing among the swarms. Under the influence of the three parts, the swarm can reach an effective and best position. In addition, the swarm is limited by $V_{max}$ when it is adjusting its own position according to the speed. The speed $V_i$ is set to be $V_{max}$ when $V_i$ exceeds $V_{max}$ [15]. Here the formal algorithm is presented and Fig. 1 shows the flowchart of PSO algorithm.

Algorithm: Standard Particle Swarm Optimizer (PSO)

1. Set the iteration number t to zero, and randomly initialize swarm S within the search space.
2. Evaluate the performance $f(x_i)$ of each particle.
3. Compare the personal best of each particle to its current performance, and set $y_i$ to the better performance, according to equation (2.3).
4. Set the global best ŷ to the position of the particle with the best performance within the swarm, according to equation (2.4).
5. Change the velocity vector for each particle, according to equation (2.1).
6. Move each particle to its new position, according to equation (2.2).
7. Let t: = t + 1.
8. Go to step 2, and repeat until convergence.

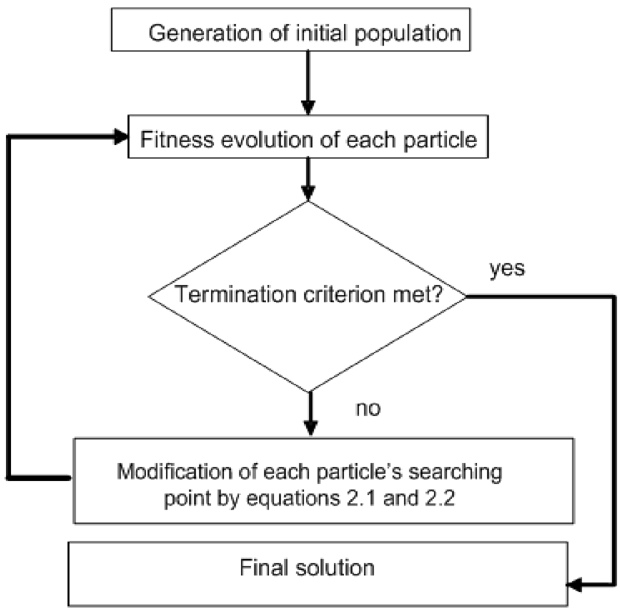


Fig. 1. Flow chart of PSO algorithm

## III. PROPOSED ALGORITHM

Three modifications have been performed on the PSO algorithm to form the new algorithm named Adaptive Particle Swarm Optimization (APSO).

### *A. First Modification*

The whole search space is divided into several segments like the net of a spider. Thus, the new algorithm generates an initial population with a uniform distribution of solutions such that every segment has solutions. By the traditional methods the initial population was created randomly where they are very much dependent on mutation operator. By dividing the whole search space into several segments –improves the search capability of the proposed algorithm instead of just relying on the use of a mutation operator. Having individuals in every segment would give better searching capability

### *B. Second Modification*

In the proposed model – the information sharing part of the swarms are made very strong. Every swarm shows interest or takes information from all other swarms that have better fitness value than its own. The swarms that have better fitness value will guide other swarms to improve their fitness value. As it is not considering only the global best solution, there is a very little chance of this model to be trapped by local optima. This idea is taken from the flocking behaviour of a spider. For instance, to minimize a function four individuals a, b, c, d is considered (Fig. 2.) and their fitness values are shown in TABLE I.

TABLE I
INDIVIDUAL AND FITNESS

| Individual | Fitness Value |
|---|---|
| a | 10 |
| b | 25 |
| c | 50 |
| d | 75 |

According to the PSO individual‘d’ should follow path p1 – which is a direct path. But As individual ‘b’ and ‘c’ has better fitness value than‘d’ – individual ‘d’ should show some interest about them. According to the proposed algorithm individual‘d’ will follow path p2. The global optimum may be somewhere in the region where ‘b’ and ‘c’ are present. The solution ‘a’ may be a local optimum.

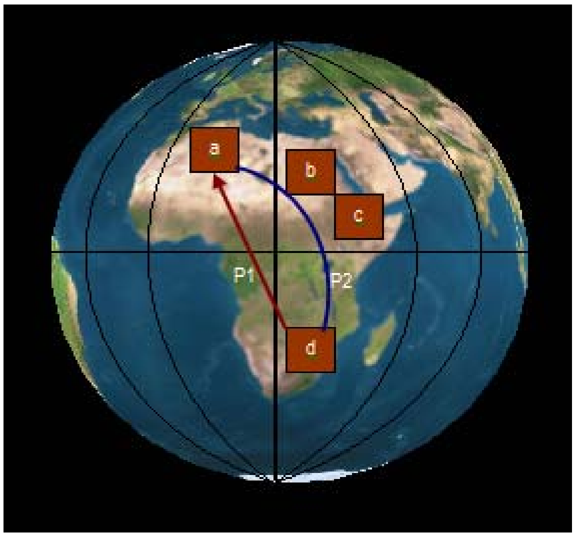


Fig. 2. Effect of Information Sharing of PSO and Modified PSO (APSO)

### *C. Third Modification*

The new algorithm uses the fitness value to adjust the learning factors ($c_1$ and $c_2$) of the swarms. In the traditional PSO, the fitness is never used. But in the proposed model - how much interest a swarm should show on others, is based on the fitness value of the swarm.

All the swarms will be ranked according to their fitness value. The swarm having smallest (in the case of function minimization) fitness value is ranked 1, and the others will be ranked in this way. When a swarm is followed by some other swarms – it shows interest according to their ranks.

### *D. The velocity update equation of Adaptive PSO:*

The velocity update equation of the APSO is as follows

Vi = Vi * wi + 1/rank(i) * rand() * ( pbest[i] – $X_i$ ) + Social_Information(i);

$X_i = X_i + V_i$;

```
Where,
Social_Information(i)
{
        posx ← 0.0
        for each individual k of the population
                if pFitness[k] is better than fitness[i]
                posx ← posx + 1 / rank(k) * rand() * ( pbest[k]- Xi );

        if(posx > Vmax) return Vmax;
        else     return posx;
}
```

Here *pFitness[k]* represents the best local fitness value seen by individual k. *fitness[i]* indicates the current fitness value of the swarm. The *Social_Information()* module gives the direction of the swarm by sharing information with all other individuals that have better fitness value. $V_{max}$ have been set with a small value to prevent a jump. The value of the *posx* may be very large. Rather than getting the value of *posx*, the proposed algorithm is more interested to get the direction from the *Social_Information()* module. The value of $V_{max}$ is automatically assigned to the swarm. If a swarm has less rank, then its $V_{max}$ would be assigned with less value, and If the swarms rank is large, then it's $V_{max}$ will be assigned with larger value. By assigning the value of $V_{max}$ in this way, the swarms are ensured to have a good local search, and can come out from the worse position very fast.

## IV. EVALUATION AND DISCUSSION

To validate the proposed algorithm, several well known multimodal functions have been chosen. The functions are difficult to optimize for any search algorithm because of their several local minima which can produce premature convergence. In all case a single global optimum exists. The selected functions have been widely studied by PSO researchers [16] [17] [18] [19] [20]. TABLE II represents the chosen benchmark functions with their properties whereas TABLE III shows the parameter for the selected test functions.

TABLE II
BENCHMARK FUNCTIONS WITH THEIR PROPERTIES

| Name | Equation | Properties |
|---|---|---|
| Sphere | $f(x)=\sum_{i=1}^{n} x_i^2$ | Smooth, strongly convex, symmetric |
| Rastrigin | $f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$ | Highly multimodal, contains millions of local optima |
| Rosenbrock | $f(x)=\sum_{i=1}^{n}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$ | Very narrow ridge makes the landscape more complicated |
| Schaffer's F6 | $f(x)=0.5-\frac{\left(\sin\sqrt{x^2+y^2}\right)^2-0.5}{\left(1.0+0.001(x^2+y^2)\right)^2}$ | Complex multimodal |
| Griewant | $f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$ | Complex multimodal |

TABLE III
PARAMETERS FOR TEST FUNCTIONS

| Name | Dimention | Initial Range | X* | f(x*) |
|---|---|---|---|---|
| Sphere | 30 | $[-100;100]^n$ | [0,0, ….0] | 0 |
| Rastrigin | 30 | $[-5.12;5.12]^n$ | [0,0, ….0] | 0 |
| Rosenbrock | 30 | $[-30;30]^n$ | [1,1, ….1] | 0 |
| Schaffer's f6 | 2 | $[-100;100]^2$ | 0,0] | |
| Griewank | 30 | $[-600;600]^n$ | [0,0, ….0] | 0 |

The inertia weight of PSO is set to 0.4 which will ensure a better convergence. The comparative results are shown in TABLE IV- VIII

TABLE IV
SPHERE'S FUNCTION

| RESULT | APSO | PSO |
|---|---|---|
| BEST | 0.00 | 0.009229 |
| WORST | 0.495870 | 20000.011719 |
| MEAN | 0.020241 | 1900.012424 |
| MEDIAN | 0.003684 | 0.012686 |
| VARIANCE | 0.000410 | 3610047.210 |
| STANDARD DEVIATION | 0.020241 | 1900.012424 |

TABLE V
RASTRIGIN'S FUNCTION

| RESULT | APSO | PSO |
|---|---|---|
| BEST | 0.00 | 49.329552 |
| WORST | 0.009889 | 392.421051 |
| MEAN | 0.003815 | 227.450577 |
| MEDIAN | 0.003540 | 248.506256 |
| VARIANCE | 0.000015 | 51733.764948 |
| STANDARD DEVIATION | 0.003815 | 222.450577 |

TABLE VI
ROSENBROCK'S FUNCTION

| RESULT | APSO | PSO |
|---|---|---|
| BEST | 0.00 | 23.0568 |
| WORST | 2301.083984 | 16186288 |
| MEAN | 1075.520323 | 21171615 |
| MEDIAN | 1016.576172 | 24.078638 |
| VARIANCE | 1156959.078 | 448237315829325 |
| STANDARD DEVIATION | 1075.620323 | 21171615.8058 |

TABLE VII
SCHAFFER'S FUNCTION

| RESULT | APSO | PSO |
|---|---|---|
| BEST | 0.00 | 0.00 |
| WORST | 0.495958 | 0.495870 |
| MEAN | 0.009918 | 0.020241 |
| MEDIAN | 0.0000 | 0.003684 |
| VARIANCE | 0.000098 | 0.000410 |
| STANDARD DEVIATION | 0.009918 | 0.020241 |

TABLE VIII
GRIEWANK'S FUNCTION

| RESULT | APSO | PSO |
|---|---|---|
| BEST | 0.00 | 0.023007 |
| WORST | 0.613968 | 0.99999 |
| MEAN | 0.024154 | 0.989998 |
| MEDIAN | 0.000006 | 0.999956 |
| VARIANCE | 0.000583 | 0.980096 |
| STANDARD DEVIATION | 0.024154 | 0.989998 |

Convergence rate of test functions for PSO and APSO are presented in Fig.3-Fig.7. From all the graphs an tables it can be easily seen that the new approach produces better solution than the traditional PSO. The proposed method gives better optima in all benchmark functions. For Sphare, Rastrigrin, Rosenbrock, Schaffer, Griewank's function APSO gives 0,0,0,0,0 optimal values respectively whereas the traditional PSO gives 0.009, 49.329, 23.056, 0.00, 0.23 values respectively. Moreover the convergence time is also less than that of traditional PSO. For example, for Griewank function the proposed method converges sharply from 2000 generations whereas the traditional PSO requires more than 10000 generations.

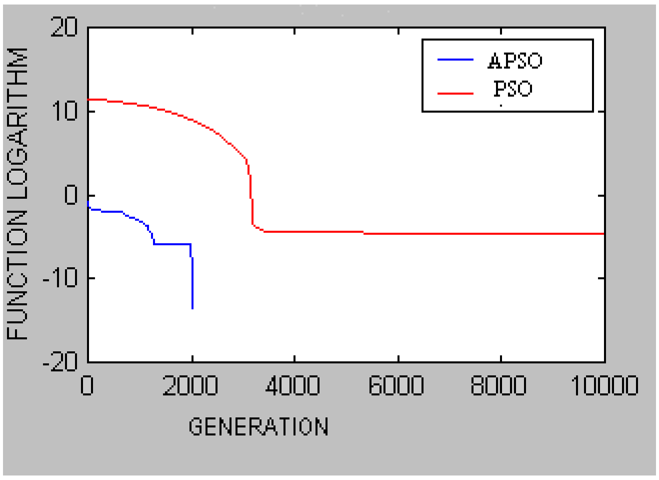


Fig. 3. Convergence of PSO and APSO on Sphere Function

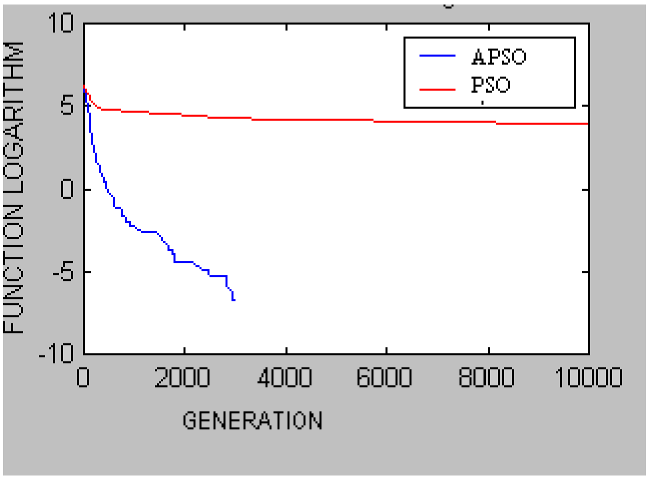


Fig. 4. Convergence of PSO and APSO on Rastrigrin Function

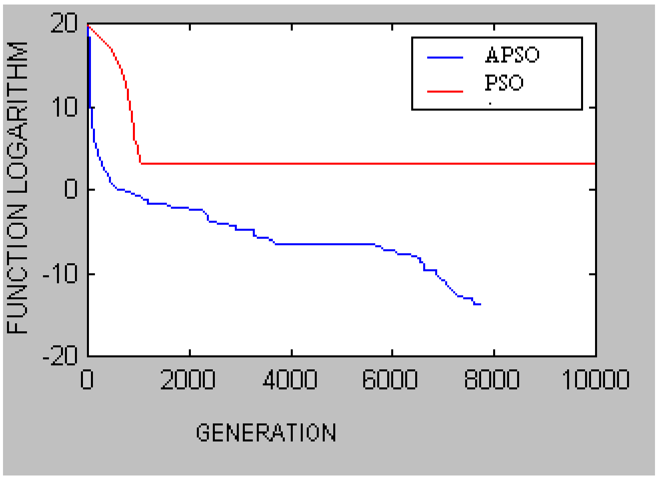


Fig. 5. Convergence of PSO and APSO on Rosenbrock Function

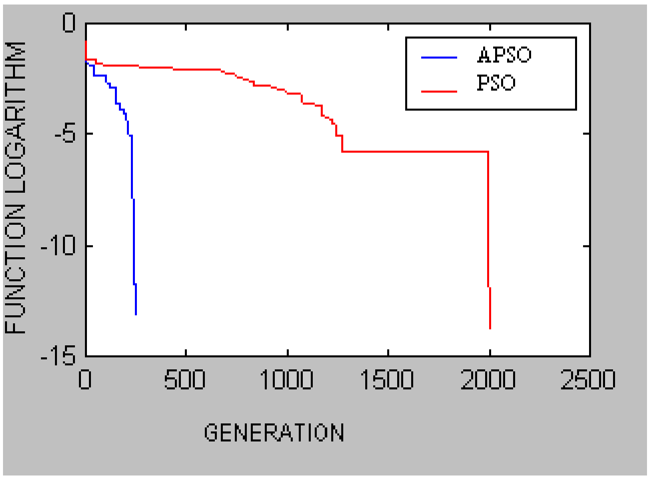


Fig. 6. Convergence of PSO and APSO on Schaffer's f6 Function

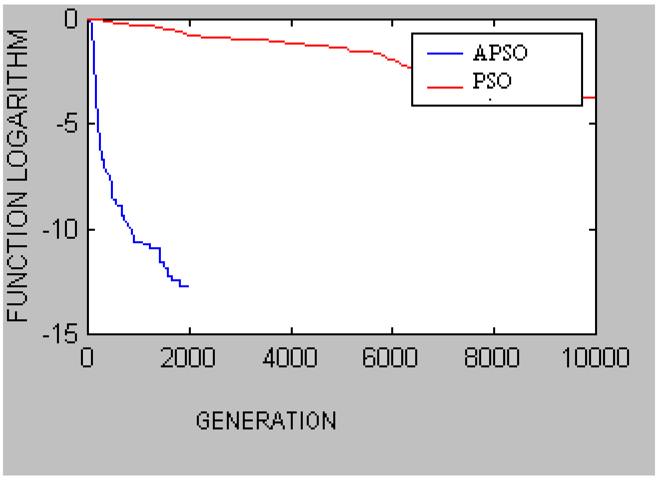


Fig. 7. Convergence of PSO and APSO on Griewank Function

# V. CONCLUSION

Computer simulations of evolution started as early as in 1954 with the work of Nils Aall Barricelli. Hans Bremermann published a series of papers in the 1960s that adopted a population of solution to optimization problems, undergoing recombination, mutation, and selection. Artificial evolution became a widely recognized optimization method as a result of the work of Ingo Rechenberg and Hans-Paul Schwefel in the 1960s and early 1970s - Rechenberg's group was able to solve complex engineering problems through evolution strategies. In this paper the authors have tried to effectively use the PSO from the pioneering work by James Kennedy, Russ Eberhart. This paper makes some modifications to the traditional PSO based on the solution space, information sharing, and learning factors to handle multimodal function optimization problems. The goal was to find a method for multimodal function optimization and test results for the benchmark function support the proposed method for the solution. The authors strongly believe that the proposed algorithm will work better for other optimization problems as well.

# REFERENCES

[1] Yong-ling Zheng, Long-hua Ma, Li-yan Zhang, Ji-xin Qian. Empirical Study of Particle Swarm Optimizer with an Increasing Inertia Weight. CEC2003, Australia.

[2] J Kennedy, and R Eberhart, “Particle swarm optimization”, Proceedings of the 4th IEEE International Conference on Neural Networks, pp-1942-1948, 1995.

[3] R Eberhart, and J Kennedy, “A new optimizer using particle swarm theory”, Proceedings of 6th International Symposium on Micro Machine and Human Science, pp.39-43, 1995.

[4] Andrew Stacey, Mirjana Jancic, Ian Grundy. Particle Swarm Optimization with Mutation, CEC2003, Australia.

[5] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, Evolutionary Programming, Vol. 7, pp. 601-610, 1998.

[6] H.Y. Fan, A modification to particle swarm optimization algorithm, Engineering Computations, Vol. 19, pp. 970-989, 2002

[7] Dipti Srinivasan and Tian Hou, Seow. Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multiobjective Optimization Problems, CEC2003, Australia

[8] Anna Hondroudakis, Joel Malard and Gregory V. Wilson, “An Introduction to Genetic Algorithms Using RPL2: The EPIC Version”, Computer Based Learning Unit, University of Leeds, 1995.

[9] Digalakis, J.G. and Margaritis, K.G., “An experimental study of benchmarking functions for genetic algorithms”, 2000 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3810-3815 vol. 5, 2000

[10] Sinclair, M.C, “The application of a genetic algorithm to trunk network routing table optimization”, 10th Performance Engineering in Telecommunications Network Teletraffic Symposium, pp. 2/1 – 2/6, 1993.

[11] Greenwood, G.W.; Lang, C.; Hurley, S, “Scheduling tasks in real time systems using Evolutionary Strategies”, Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems, pp. 195-196, 1995.

[12] Fogel, D. and Sebald, A.V., “Use Of Evolutionary Programming In The Design Of Neural Networks For Artifact Detection”, Proceedings of the Twelfth Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1408-1409, 1990.

[13] Meshref, H. and VanLandingham, H., “Artificial immune systems: application to autonomous agents”, 2000 IEEE International Conference on Systems, Man, and Cybernetics, pp. 61-66 vol. 1, 2000.

[14] A. R. Cockshott and B.E. Hartman. Improving the fermentation medium for Echinocandin B production part ii: Particle swarm optimization. Process Bio-chemistry, 36: 661-669, 2001.

[15] P.C. Fourie and A.A. Groenwold. The particle swarm optimization algorithm in size and shape optimization. Struct, Multidisc. Optim., 23: 259-267, 2002.

[16] Battiti, R., and Tecchioli, G., The Reactive Tabu Search. ORSA Journal on Computing, Vol. 6, 126-140, 1994.

[17] Davis., L., Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991.

[18] Woodruff, D.L., and Zemel, E., Hashing vectors for tabu search. Annals of Operations Research, 41, 123-137, 1993.

[19] N. Higashi and I. H. Particle swarm optimization with Gaussian mutation. In Swarm Intelligence Symposium (Indianapolis, Indiana), pages 72-79, Piscataway, NJ., April 2003. IEEE Service Center.

[20] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In Congress on Evolutionary Computation (Honolulu, Hawaii), Piscataway, NJ., May 2002. IEEE Service Center.

[21] Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. In International Conference on Evolutionary Computation (Anchrage, AK), Piscataway, NY., 1998. IEEE Service Center.

[22] F. van den Bergh. An Analysis of Particle Swarm Optimization. PhD thesis, Facaulty of Natural and Agricultural Science, University of Petroria, South Africa, November 2002.

[23] Lothar M. Sumitt. Theory Of Genetic Algorithms. Technical Report 2000-2-002; September 23, 2000