IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 6, Dec -Jan, 2016
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

A Comparison Of Particle Swarm Optimization Algorithm With Genetic Algorithm On Benchmark Functions

Ashok Kumar[1], Brajesh Kumar Sing[2], B.D.K.Patro[3]
[1,2]Computer Science & Engineering Department, DR. A.P.J.Abdul Kalam Technical University, Lucknow, India
[3]R.B.S.Engineering & Technical Campus, Bichpuri, Agra-283105, India

Abstract: Optimization of one or more objective functions is a requirement for many real life problems. Due to their wide applicability in business, engineering and other areas, many algorithms have developed to solve these problems to get optimal solutions in minimum possible time. Particle Swarm Optimization is a very simple and popular optimization algorithm. Many Researchers are using PSO in their optimization problems. Genetic algorithm (GA) is also used in optimization problems. it is more difficult to apply in real life problems. In this paper we generate dataset of PSO and compare with data set of GA available for noisefree BBOB testbed. Particle swarm optimization (PSO) shows superior performance in several real-world applications. Comparision of the performance of PSO with GA is done on COCO framework software. This comparison is performed on 24 noise-free functions.

Keywords: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), COCO.

1. INTRODUCTION

Many real life problems can be formulated as optimization problems. The objective of an optimization problem is to find out maximum or minimum value of an objective function. Generally optimization problems are complex and hard to solve and thus the algorithms solving such problems becomes important. A large number of randomized algorithms like evolutionary approach and swarm intelligence algorithms, Differential Evolution and Particle Swarm Optimization [3] have been proposed in literature to solve these problems. Out of these algorithms PSO has been immensely researched due to its simplicity and potential to solve any optimization problem.

Genetic algorithms are adaptive heuristic search algorithm premised on Darwin's evolutionary ideas of natural selection and genetic[14]. Genetic algorithm is used in computational models developed by Holland who is inspired by evolution **[1, 2].** This algorithm encodes a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information.GA is often viewed as function optimizer, although the range of problems to which GA has been applied is quite broad.

Particle swarm optimization (PSO) is also an evolutionary computational model which is based on swarm intelligence. PSO is developed by Kennedy and Eberhart who have been inspired by the research of the artificial life **[5].** Similar to GA, PSO is also an optimizer based on population. The system is initialized firstly in a set of randomly generated potential solutions, and then performs the search for the optimum solution iteratively. The PSO does not possess the crossover and mutation processes adopted in GA. It finds the optimum solution by swarms following the best particle. Compared to GA, the PSO has much more profound intelligent background and could be performed more easily. Due to its advantages, PSO is not only suitable for scientific research, but also engineering applications. Presently the PSO has attracted broad attention in the fields of evolutionary computing, optimization and many others [6-10]. The PSO has been applied widely in the function optimization, artificial neural networks' training, pattern recognition, fuzzy control and some other fields. Some improved PSO algorithms have been developed.

The paper is organized as follows. In **Section 2** we introduce GA and PSO methods used in the study. **Section 3** we explain about experimental setup of COCO framework, benchmark functions on which compare the results and PSO setting used in PSO Algorithm. The experimental results and discussion is presented on 3 and 5 Dimension **in Section 4**. Finally **Section 5** contains a conclusion focuses on future enhancement in PSO algorithm.

2. METHODS

2.1 Genetic Algorithm

For a specific problem, the **GA** codes a solution as an individual chromosome. It then defines an initial population of those individuals that represent parts of feasible solutions of the problem. The search space therefore, is defined as the solution space in which each feasible solution is represented by a distinct chromosome. Before the search starts, a set of

chromosomes is randomly chosen from the search space to form the initial population. Next, through computations the individuals are selected in a competitive manner, based on their fitness measured by a specific objective function. The genetic search operators such as selection, mutation and crossover are then applied in sequence to obtain a new generation of chromosomes in which the expected quality over all the chromosomes is better than that of the previous generation. This process is repeated until the termination criterion is met, and the best chromosome of the last generation is reported as the final solution.

Algorithm for Genetic Algorithm [4]:

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete

3.1 [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

3.2 [Crossover] with a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

3.3 [Mutation] with a mutation probability mutate new offspring at each locus (position in chromosome).

3.4 [Accepting] Place new offspring in a new population

4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition (for example number of populations or improvement of the best solution) is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

2.2 PSO Algorithm

The particle i of the swarm can be represented by an d-dimensional Vector Xi = (xi1, xi2, …, xid). The velocity of this particle can be represented by another d-dimensional vector Vi = (vi1, vi2, …., vid). The fitness of each particle can be evaluated according to the objective function of optimization problem. The best previously visited position of the particle i is noted as its individual best position Pi = (pi1, pi2, …, pid). The position of the best individual of the whole swarm is noted as the global best position G = (g1, g2, …, gd). At each step, the velocity of particle and its new position will be assigned according to the following two equations:

$$v_{id}^{t+1} = \omega * v_{id}^{t} + c_1 r_1 (P_{id} - x_{id}) + c_2 r_2 (G_{best} - x_{id}^{t}) \quad (1)$$

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

Where,

x_{id} = position vector of particle i in d-dimension.
v_{id} = velocity vector of particle i in d-dimension.
P_{id} = personal best
G_{best} = global best
ω = the inertia weight
r_1, r_2= independently uniformly distributed random variables with range (0, 1).
c_1, c_2 = positive constant parameters called acceleration coefficients.

At each step of the iteration, all particles move to a new place and the direction and distance are defined by their velocities. Equation (1) shows that the velocity of any given particle is a stochastic variable and that it is prone to create an uncontrolled trajectory, allowing the particle to follow wider cycles in the design space, as well as letting even more particles to escape it. In order to limit the impact of this phenomenon, velocity should be clamped into a reasonable interval. Here the new constant v_{max} is defined which represent the maximum value of velocity:

If $v > v_{max}$, then $v = v_{max}$ (3)

If $v_{i,j} < -v_{max}$, then $v = -v_{max}$ (4)

Normally, the value of vmax is set empirically, according to the characteristics of the problem.

3. EXPERIMENTS

A. Experimental Setup and Data Sampling

COCO (**Co**mparing **Co**ntinuous Optimizers) [10] is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds, experimentation templates which are easy to parallelize, and tools for processing and visualizing data generated by one or several optimizers. the COCO platform has been used for Black-Box-Optimization-Benchmarking(BBOB) workshops.

To check the performance of PSO algorithm with GA dataset, BBOB noiseless test-bed is used. Optimal value of particles position is searched in closed interval –5 to +5. BBOB has 24 benchmark functions and they are checked for six dimensions. The population size is kept 50 and total number of function's evaluation is made dependent on the dimension and it varies

as the dimension changes. To calculate the function evaluation formula dimension* 10^6 is used. The total function evaluation is calculated by the formula i.e. ***Total function evaluation =Maximum function evaluation * Function evaluation in one iteration.*** The data set which is used to compare the performance of the proposed algorithm has been taken from the COCO framework [11].

B. Benchmark Functions:

24 noise free real-parameter single-objective benchmark functions are presented. These functions are classified in five groups include separable functions (f1 - f5), functions with low or moderate conditioning (f6 - f9), unimodal functions with high conditioning (f10 - f15), multi-modal functions with adequate global structure (f16 - f19), and multi-modal functions with weak global structure (f20 - f24) as listed in Table – I. Our intention behind the Selection of benchmark functions was to evaluate the performance of algorithms with regard to typical difficulties which we believe occur in continuous domain search. All benchmark functions are scalable with the dimension. Most functions have no specific value of their optimal solution (they are randomly shifted in x-space). All functions have an artificially chosen optimal function value (they are randomly shifted in f-space).

C. PSO Settings

PSO has several parameters. the number of particles in the swarm(swarm size), maximum velocity (v_{max}),the parameters for attraction towards personal best and the neighborhoods best found solutions (c_1 and c_2) and the inertia weight (w).for PSO we used these settings: swarm size==40,c_1 and c_2 = 1.4944 and w= 0.792.

TABLE I. **BBOB BENCHMARK FUNCTIONS**

Group	f#	Function Name
Separable	f1	Sphere
	f2	Ellipsoidal
	f3	Rastrigin
	f4	Büche-Rastrigin
	f5	Linear Slope
Low or moderate conditioning	f6	Attractive Sector
	f7	Step Ellipsoidal
	f8	Rosenbrock, original
	f9	Rosenbrock, rotated
Unimodal with high conditioning	f10	Ellipsoidal
	f11	Discus
	f12	Bent Cigar
	f13	Sharp Ridge
	f14	Different Powers
Multi-modal with adequate global structure	f15	Rastrigin
	f16	Weierstrass
	f17	Schaffers F7
	f18	Schaffers F7, moderately ill-conditioned
	f19	Composite Griewank-Rosenbrock F8F2
Multi-modal with weak global structure	f20	Schwefel
	f21	Gallagher's Gaussian 101-me Peaks
	f22	Gallagher's Gaussian 21-hi Peaks
	f23	Katsuura
	f24	Lunacek bi-Rastrigin

4. RESULTS AND DISCUSSION

Performance of PSO and GA algorithms is shown in Figure number 1-2 on Dimension Size 3 and 5. Results from experiments according to [11] on the benchmark functions given in [12, 13] are presented in figure number 1-2.

From Figure 1, it is clearly observed that on 3-D the performance of PSO has improved over GA on separate, Moderate and ill-conditioned functions. While GA does not perform on multi-modal and weakly structured multi modal

functions. Overall the PSO gives better results on 3-Dimension search space.

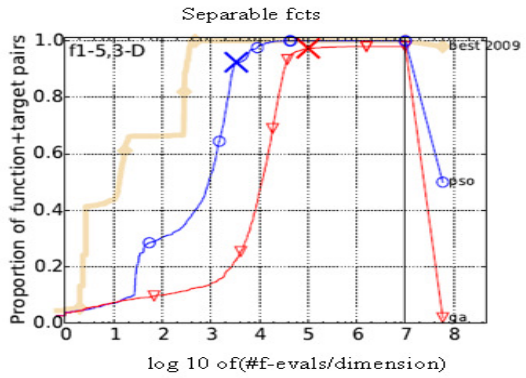
From Figure 2, it is clearly observed that on 5-D the performance of PSO has improved over GA on moderate and ill-conditioned functions. However, its convergence rate has decreased in separate functions. While GA does not perform on multi-modal and weakly structured multi modal functions on 5-Dimension search space.

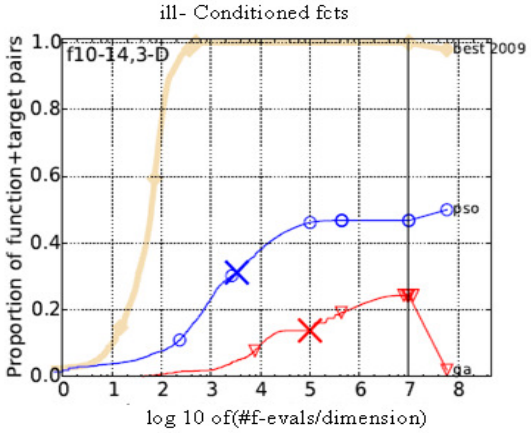
The performance of PSO is outstanding in comparison to the GA algorithm tested on COCO framework software used for the BBOB workshops. This comparison is performed on 24 noise-free functions. it is simple, robust, converges fast, and finds the optimum in every run. in addition, it has few parameters to set, and same settings can be used for many different problems.

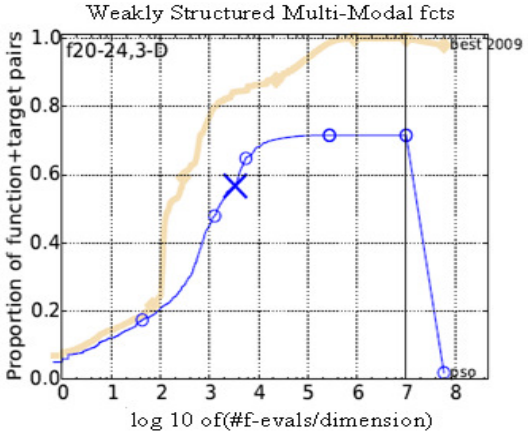
From the results it is clear that on dimension 3 the PSO performed well over GA. In 5 dimensions, GA has dominated PSO in Separable functions. However its convergence rate has decreased in separable functions. The performance of PSO is good.

5. CONCLUSION

Proposed PSO is performing in better way as compare genetic algorithm .many researchers developed different variants of PSO by tuning its parameters. Comparison results confirm that updated velocity help swarm to recover from local optima and prevent premature convergence. PSO performs consistently better in separate, moderate and ill-conditioned functions in 3-5 dimensions. There is further need to improve PSO algorithm so that it can perform well on multi-model and weakly structured multi-modal type problems. The proposed algorithm PSO outperforms with GA for nearly all 24 benchmark functions of different category provided by BBOB 2009.







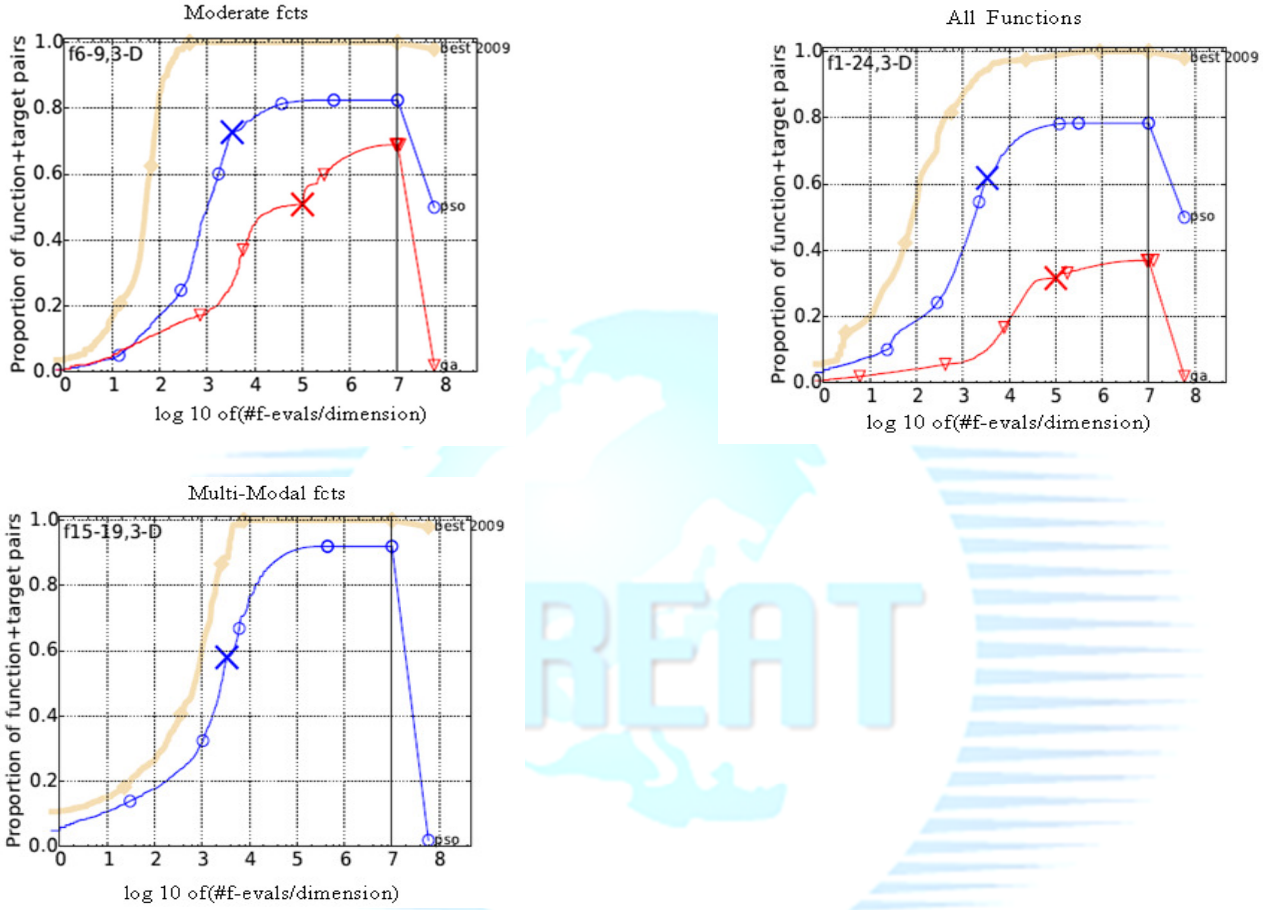


Figure 1: Empirical cumulative distribution of number of objective function evaluations divided by dimension for PSO and GA Algorithm on 3-D.

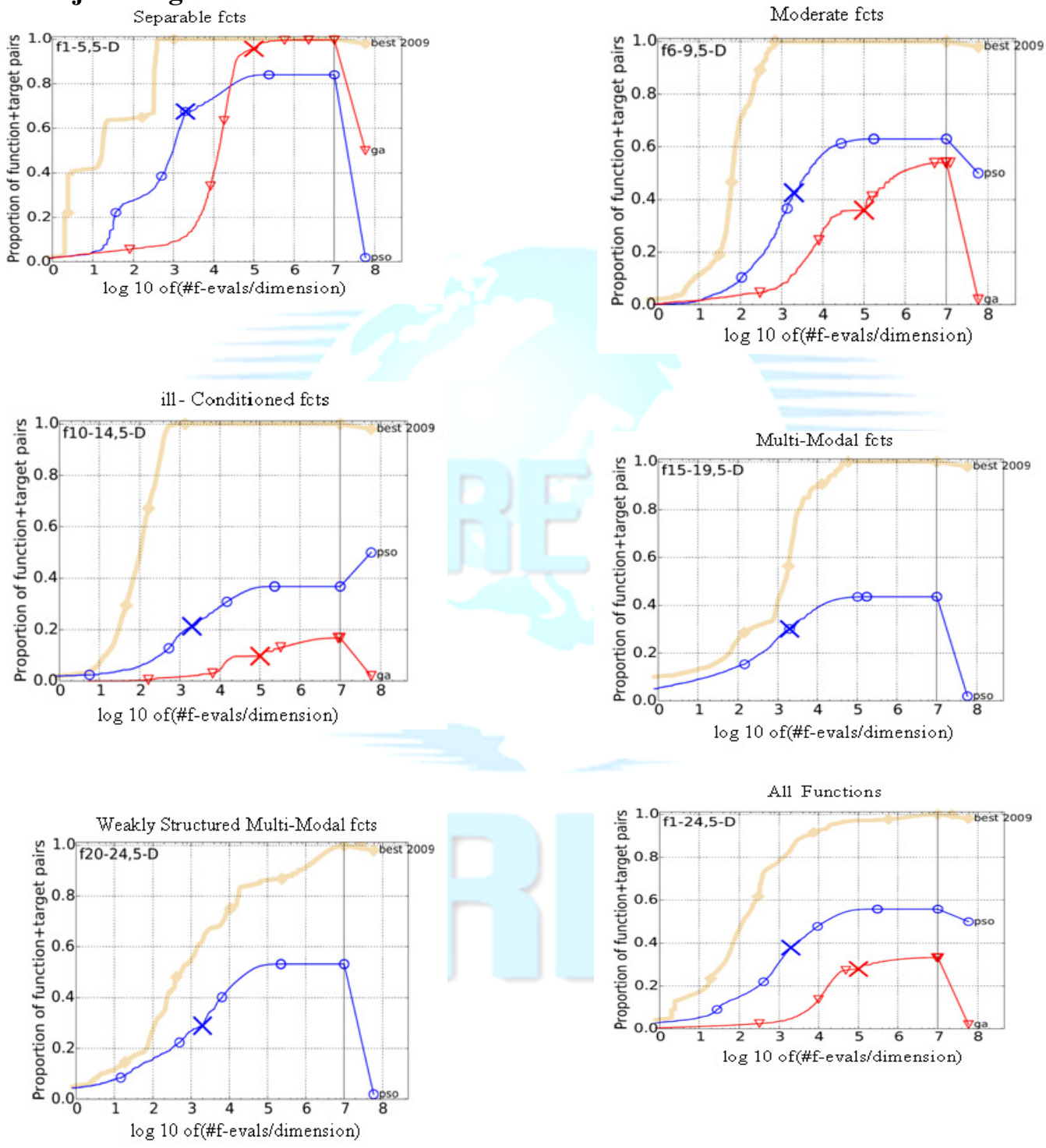


Figure 2: Empirical cumulative distribution of number of objective function evaluations divided by dimension for PSO and GA Algorithm on 5-D.

References:

[1] J.H. Holland, Adaptation in Natural and Artificial System, the University of Michigan Press, Ann Arbor, 1975.

[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning. Reading, MA: Addison-Wesley, 1989.

[3] J. Kennedy and R. Eberhart, *"Particle Swarm Optimization"*, IEEE International Conference on Neural Networks, vol.4, pp. 1942-1948, Nov/Dec 1995.

[4] http://www.obitko.com/tutorials/genetic-algorithms/index.php.

[5] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, Sixth International Symposium on Micro Machine and Human Science, pp. 39-43,1995.

[6] J. Kennedy the PSO: social adaptation of knowledge, Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, Indiana, pp.303-310, 1997.

[7] Maurice Clerc and James Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, Vol. 6, pp. 58-73, 2002.

[8] Ioan Cristian Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, Vol. 85, pp. 317-325.2003.

[9] Shigenori Naka, Takamu Genji, Toshiki Yura, and Yoshikazu Fukuyama. A hybrid panicle swarm optimization for distribution state estimation, IEEE Transactions on Power Systems, Vol. 18, pp.60-68, 2003

[10] http://coco.gforge.inria.fr/.

[11] Hansen, Nikolaus, et al. "Real-parameter black-box optimization benchmarking 2010: Experimental setup." (2010).

[12] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012:Experimental setup.Technical report, INRIA,2012.

[13] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization.benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[14] Goldberg, D.E. , 1989. "Genetic algorithms in search, optimization, and machine learning "Reading, MA: Addison-Wesley .