International Journal For Technological Research In Engineering
Volume 6, Issue 9, May-2019 *ISSN (Online): 2347 - 4718*

PERFORMANCE EVALUATION FOR ADAPTIVE PARTICLE SWARM OPTIMIZATION OVER CLASSICAL PARTICLE SWARM OPTIMIZATION

Ashish Garg[1], Nitin Halkara[2]
[1]M. Tech Student, [2]Assistant Professor,
Shekhawati Institute of Engg & Technology, Sikar, India

Abstract: The application of the Particle Swarm Optimization (PSO) algorithm successfully applied to wide range of engineering problems. In very recent this algorithm becomes very popular due to its simplicity and effectiveness. Researchers have explored PSO algorithm and made it adaptive with dynamic problem and these modified algorithm called as Adaptive Particle Swarm Optimization (APSO) algorithm. APSO algorithm is explored in this paper. The choice of this algorithm is made over Classical Particle Swarm Optimization (CPSO) and this is explained for different benchmark functions as Test functions. In order to find out global minima APSO showed very good performance for these Test functions. This algorithm can be successfully applied to various digital communication systems for improving performance.
Key words: Test Function; Particle swarm optimization; Adaptive particle swarm optimization;

I. INTRODUCTION

Several evolutionary algorithms have emerged in the past years from biological entities behavior and evolution. Darwin's theory of evolution is natural selection is inspiration source of for EAs. EAs are widely used for the solution of single and multi-objective optimization problems. Swarm Intelligence (SI) algorithms are also a special type of EAs. SI can be defined as the collective behavior of decentralized and self organized swarms. SI algorithms among others include Particle Swarm Optimization (PSO) [1] and Ant Colony Optimization [2].

PSO is an evolutionary optimization algorithm that is formed form the swarm behavior of bird flocking and fish schooling [3]. PSO is an easy to implement with less computational complexity. Particle Swarm Optimization (PSO) is a powerful method of optimization that has been widely used for solving different optimization problems. It is widely used to find the global optimum solution in a complex search space. In this paper discuss on theoretical as well as detailed explanation of the PSO algorithm. Apart from this, advantages and disadvantages, the effects of the various parameters have been discussed. Finally, this dissertation presents improved version of PSO also. In section 2, different benchmark functions as test problems are selected for performance evolution and mathematical modeling of these functions are described.

II. CLASSICAL PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was introduced by Kennedy and Eberhart in 1995. This optimization algorithm is evolutionary computation technique. This modem algorithm is very effective to solve global optimization problems [4]. It starts with random initialized population or swarm and population will be modified in search of optimal solution. Population of random solutions in swarm is defined as a "particle". Initially, every particle flies into a search space. Then each particle moves as per flying experiences of itself and its neighborhoods. At [illegible] iteration, position of particles is updating to find global optimal solution. In comparison of other optimization methods, PSO is simple to implement and fast convergence response [5]. PSO algorithm consists of 'n' particles in swarm that represents candidate solution. Each particle is represented by its current position 'x' and current velocity 'v' in 'd'-dimensional space. Wherever it is iterative process, in every iteration the position is getting updated based on best known individual position as well as best known swarm position.

iter	Iteration number
i	Particle index
d	Dimension
v_i^d	Velocity of i^{th} particle in d^{th} dimension
x_i^d	i^{th} Particle position in d^{th} dimension
c_1, c_2	Acceleration constants
R_1, R_2	Random numbers with uniform distribution [0, 1]
$gbest^d$	Swarm global best position in d^{th} dimension
$pbest^d$	Particle best position of i^{th} particle in d^{th} dimension

Where x_i^d and v_i^d are position and velocity for i^{th} particle in d^{th} dimension respectively. $pbest^d$ is particle individual best known position and $gbest^d$ is swarm global best known position. c_1 and c_2 are acceleration constants. R_1 and R_2 are random numbers in range of [0, 1].

III. PSO ALGORITHM PARAMETERS

PSO parameters may affect performance for optimization. For any given optimization problem, PSO parameter's values have large impact on the efficiency of the PSO method. The basic PSO parameters are swarm size or number of particles, iterations, velocities, and acceleration coefficients.

Swarm size

Swarm size is the number of particles in the swarm. A bigger swarm size may reduce the number of iterations need to obtain a good optimization result but it may increase the computational complexity per iteration as well as consume more time.

Iterations

The number of iterations also play important role to obtain a good result. Low number of iterations may stop the search process prematurely and large number of iterations may add computational complexity and time consuming.

Velocity Components

The velocity components are also very important for updating particle's velocity. Particle velocity vd is confined between −vdand vd . If vd is too big then solution is away from global best solution and if it is too small then solution may not reach to beat solution. Acceleration coefficients The acceleration coefficients c1 and c2, together with the random values, maintain the stochastic influence of the cognitive and social components of the particle's velocity respectively. In general fixed values of acceleration coefficient are being considered, but wrong initialization of may result in diverse from optimum point. Due to fast convergence, simplicity and easy implementation, PSO widely adapted for optimization in different fields.

ADVANTAGES AND DISADVATAGES OF CPSO

PSO algorithm is the one of the most powerful methods for solving the complex global optimization problems while there are some disadvantages of the PSO algorithm. The advantages and disadvantages of PSO are discussed below:

Advantages

PSO is a simple algorithm to implement. It can be applied both in scientific research and engineering problems. In PSO algorithm, there are very few parameters in compare to other optimization techniques. PSO gives fast convergence.

Disadvantages

Wrong values of PSO parameters may diverse the results from optimum or may result slow convergence.

IV. ADAPTIVE PARTICLE SWARM OPTIMIZATION

This algorithm was proposed by Arumugam and Rao [6]. Five principles of swarm intelligence have been discussed by them:

1. proximity principle
2. diverse response principle
3. quality principle
4. stability principle
5. adaptability principle

This algorithm considered global and local best fitness values into account, satisfying the above principles also. The modification in inertia is done as: is the average of local best fitness values. After iteration, inertia weight is gets updated with global best and local best fitness values. More the difference between global and local fitness values, result a larger value of inertia weight. A larger value of the ratio implies that the average best fitness value of particles nears the global best value and hence calls for a more intensive search by reducing the inertia weight of the particles. So it may result a dynamic velocity of particles. Reason of selecting 1.1 is that the inertia weight never becomes to zero in case the global best fitness value equals the average value. This inertia weight is then incorporated in the velocity update equation, as shown below:

Where w is inertia weight.

Flowchart of APSO is mentioned below in Fig. 1.

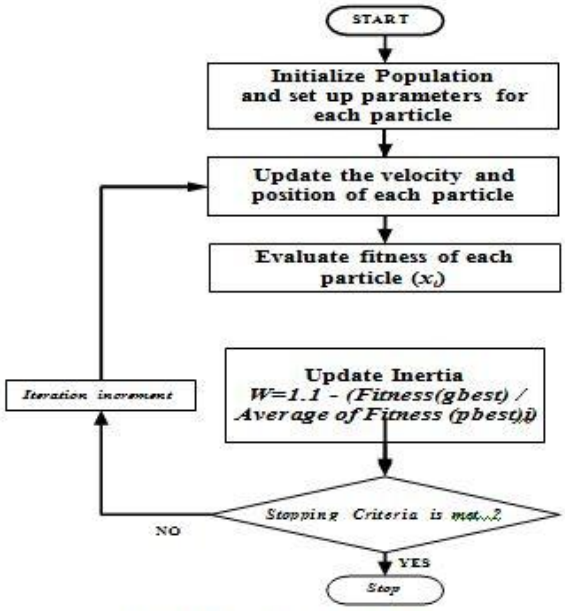

Fig. 1 Flowchart of APSO

V. TEST FUNCTIONS

To compare the implemented PSO variants, different optimization benchmark problems have been selected based on their use in the literature. PSO variants that perform well for the optimization benchmark problems will likely perform well for other optimization problems in digital communication systems. For performance justification of any new optimization, it is essential to validate its performance and compare with other existing algorithms over different types of test functions.

Test functions are very important for validating and comparing the performance of optimization algorithms. Test functions selected for present study have diverse properties so that optimization algorithms performance can be examined fairly. For this purpose, we have reviewed and compiled 8 different benchmark functions. Different optimization test problems have been implemented and are used to compare different PSO and proposed APSO variant. Test problems have been implemented, including the Ackley test problem Equation (5), bird test function Equation (6), Giunta test problem Equation (7), the Griewank test problem Equation (8) and Rastrigin test problem Equation (9).

Ackley test function

The Ackley test function is widely used for testing optimization algorithms. Its plot shows that it is characterize by a nearly flat outer region, and a large valley at the centre. In this test function for optimization algorithms, there is probability to be trapped in one of its many local minima.

The Ackley test function has the following objective function [7]:

Where a=-20, b=0.2, c=2*pi and d=2.

The topography of Ackley for two dimensions is shown in Fig. 2.

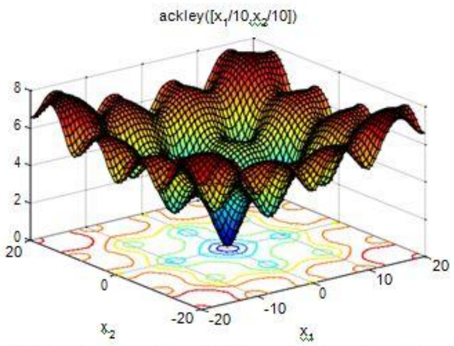

Fig. 2 The topography of Ackley for two dimensions

Search space for defined problem is,
The global minimum is at f(x1, x2) = f(3, 0.5) = 0.
$-20<x_1<20$
$-20<x_2<20$

Bird test function

This function is based on differentiable, non-scalable and multi-model.

The topography of bird test function for two dimensions is shown in Fig. 3.

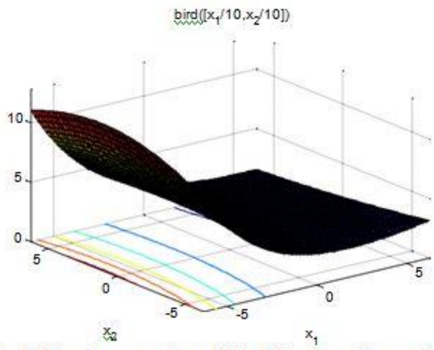

Fig. 3 The topography of Bird for two dimensions

Search space for defined problem is,

$-2\pi<x_1<2\pi$

$-2\pi<x_2<2\pi$

The global minimum is located at (−1.58214, −3.13024) is −106.764537.

Giunta Test Function

Giunta Test Function is Continuous; Differentiable and Multimodal bases test function. The topography of Giunta for two dimensions is shown in Fig. 4.

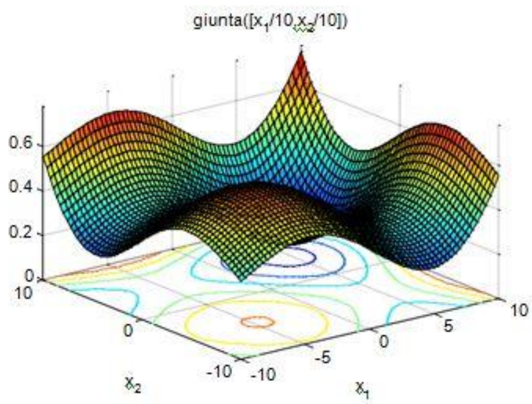

Fig. 4 The topography of Giunta for two dimensions

Search space for defined problem is,
$-10<x_1<10$
$-10<x_2<10$
The global minima at (0.46732,0.46732) is 0.064470.

Griewank test function

The Griewank function has many local minima within regularly distributed bounded search space. The topography of Griewank for two dimensions is shown in Fig. 5.

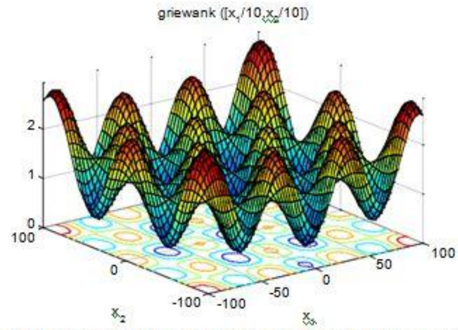

Fig. 5 The topography of Griewank for two dimensions

Search space for defined problem is,

$-100 < (x_1, x_2) < 100$

The global minimum is at f(x1, x2) = f(0, 0) is 0.

V. SIMULATION RESULTS

In this paper, the proposed APSO optimization algorithm is applied to solve the different test functions and compared with CPSO optimization results. There are 5 different test functions selected as test function to justify the performance of proposed optimization algorithm. The simulations are carried out using Matlab simulation tool. Different parameters for CPSO and APSO selected for present study is shown in appendix. Maximum iterations for both algorithms are 200 and initial swarm is randomly generated. There are 10 different runs have been conducted for each test function.

Test Function 1 (Ackley):

Fig. 6 shows the iteration of each algorithm to hit 1% tolerance band of best fitness value, e.g. for test function 1, 0 is best fitness value and from starting point to achieve 0.01 to 0 value, it takes 106 iteration. To find out best fitness value, each run does not assure to achieve same results due to stochastic nature of optimization algorithms.

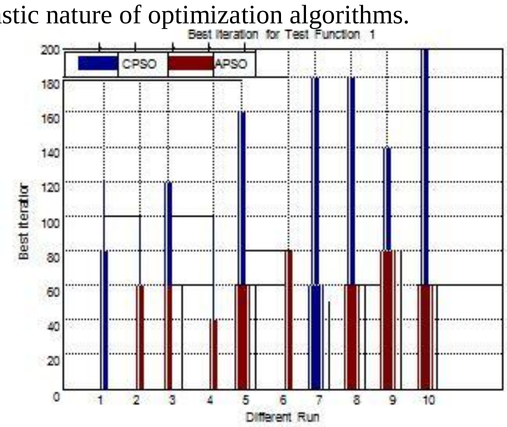

Fig. 6 Iteration to enter 1% tolerance band of global minima for CPSO and APSO

Table 1 shows the position in two dimensional area for best fitness value or minima find out by optimization algorithm and in this table best fitness value find out by CPSO and APSO are shown as well as corresponding positions of these.

Table- 1 Best fitness value and corresponding positions for CPSO and APSO for Test Function 1

TF 1	CPSO			APSO		
	Position		Best Fitness value	Position		Best Fitness value
	x_1	x_2	f(x)	x_1	x_2	f(x)
Run 1	-0.00039	-0.00213	0.006280	0.00042	-0.00042	0.00170
Run 2	0.00145	-0.00178	0.006655	0	0	0
Run 3	0.00108	0.000716	0.003713	0	0	0
Run 4	-0.00020	-0.000898	0.002625	0	0	0
Run 5	-0.00190	-0.000889	0.006075	0	0	0
Run 6	-0.002375	0.001354	0.007932	0	0	0
Run 7	0	0.003360	0.009808	0	0	0
Run 8	0.001018	-0.000542	0.003299	0	0	0
Run 9	-0.00057	-0.002103	0.006292	0	0	0
Run 10	0.004002	0.0015434	0.012622	0	0	0

Fig. 7 shows the mean of fitness of each particle in swarm, in present study swarm size is 200, means 200 particle have different fitness, so mean fitness is,

$f(x)_{mean} = {}^{200} f(x)_i$ $_{i=1}$

Fig. 8 shows the best fitness of in swarm with respect to iteration for CPSO and APSO.

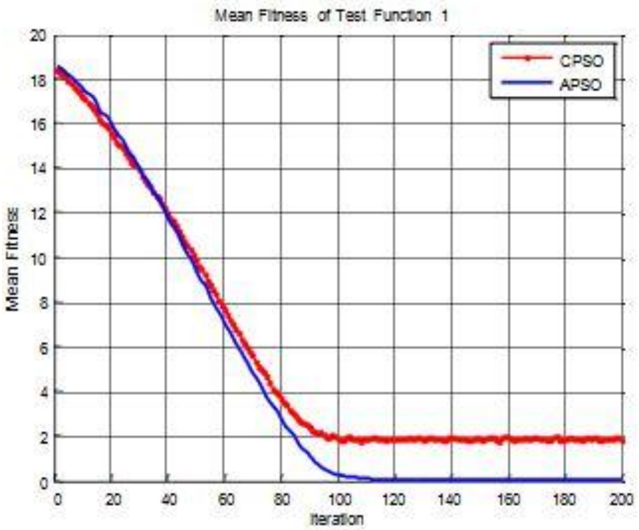

Fig. 7 Mean of different fitness of every iteration for CPSO and APSO

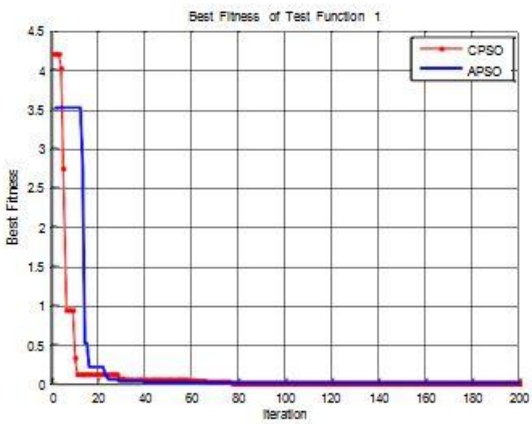

Fig. 8 Best fitness at each iteration for CPSO and APSO

Test Function 2 (Bird):

Fig. 9 shows the iteration of each algorithm to hit 1% tolerance band of best fitness value for 10 different runs.

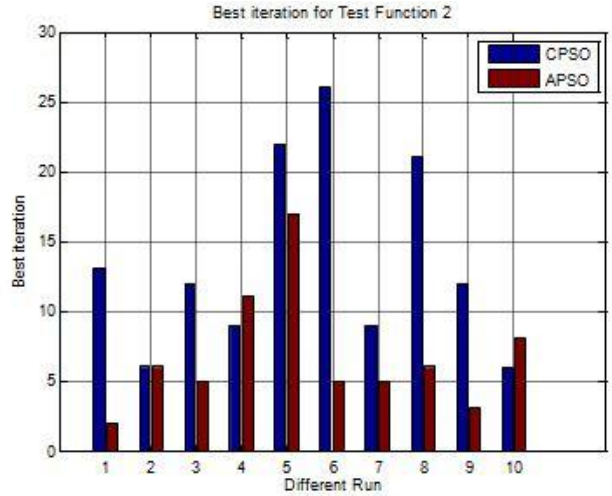

Fig. 9 Iteration to enter 1% tolerance band of global minima for CPSO and APSO for Test Function 2

Table 2 shows the numerical representation of iteration to enter in 1% band in global minima for Test Function 2.

Table 2 shows the position in two dimensional area for best fitness value or minima for test function 2 to find out by optimization algorithm and in this table best fitness value find out by CPSO and APSO are shown as well as corresponding positions of these.

Table- 2 Best fitness value and corresponding positions for CPSO and APSO for Test Function 2

TF 2	CPSO			APSO		
	Position		Best Fitness value	Position		Best Fitness value
	x_1	x_2	f(x)	x_1	x_2	f(x)
Run 1	-1.58217	-3.12964	-106.76441	-1.58214	-3.13024	-106.76453
Run 2	-1.58382	-3.13135	-106.76398	4.70104	3.15293	-106.76453
Run 3	4.70232	3.15171	-106.76410	-1.5821	-3.13025	-106.76453
Run 4	-1.58302	-3.12999	-106.76442	4.69446	3.15667	-106.75663
Run 5	4.70063	3.15255	-106.76449	4.70104	3.15293	-106.76453
Run 6	-1.58127	-3.13097	-106.76435	4.70104	3.15293	-106.76453
Run 7	-1.58251	-3.13051	-106.76450	-1.58214	-3.13024	-106.76453
Run 8	4.70226	3.15473	-106.76389	4.70414	3.15320	-106.76320
Run 9	-1.58301	-3.13013	-106.76443	4.70103	3.15296	-106.76453
Run 10	4.70064	3.15402	-106.76435	4.70104	3.15293	-106.764536

Fig. 10 shows the mean fitness and Fig. 11 shows the best fitness with respect to iteration for CPSO and APSO for Test Function 2.

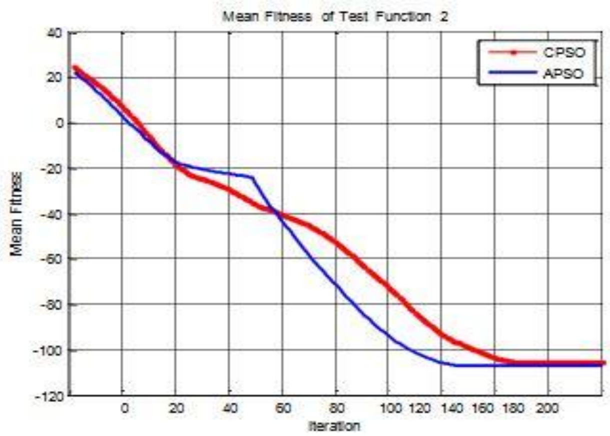

Fig. 10 Mean of different fitness at each iteration for CPSO and APSO for Test Function 2

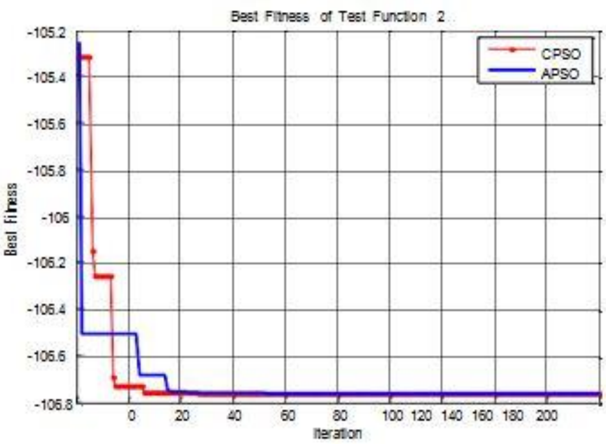

Fig. 11 Best fitness at each iteration for CPSO and APSO for Test Function 2

Test Function 3 (Giunta):

Fig. 12 shows the iteration of each algorithm to hit 1% tolerance band of best fitness value for 10 different runs.

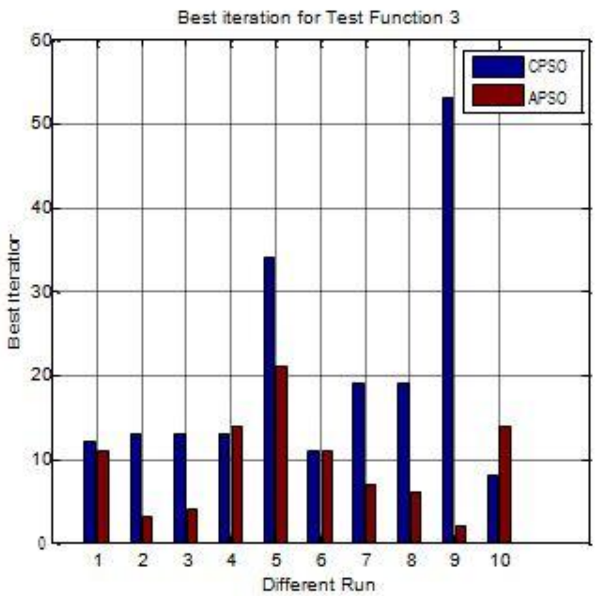

Fig. 12 Iteration to enter 1% tolerance band of global minima for CPSO and APSO for Test Function 3

Table 3 shows the position in two dimensional area for best fitness value or minima for test function 3 to find out by optimization algorithm and in this table best fitness value find out by CPSO and APSO are shown as well as corresponding positions of these.

Table- 3 Best fitness value and corresponding positions for CPSO and APSO for Test Function 3

TF 3	CPSO			APSO		
	Position		Best Fitness value	Position		Best Fitness value
	x_1	x_2	f(x)	x_1	x_2	f(x)
Run 1	0.4672569	0.4673036	0.06447042	0.467238	0.46728	0.06447043
Run 2	0.4673337	0.4673587	0.06447042	0.467320	0.46732	0.06447042
Run 3	0.4673552	0.4674029	0.06447042	0.467320	0.46732	0.06447042
Run 4	0.4672804	0.4674181	0.06447043	0.467320	0.46732	0.06447042
Run 5	0.4672119	0.4674182	0.06447044	0.467320	0.46732	0.06447042
Run 6	0.4672721	0.4674558	0.06447044	0.467320	0.46732	0.06447042
Run 7	0.4673244	0.4673028	0.06447042	0.467320	0.46732	0.06447042
Run 8	0.4672764	0.4673466	0.06447042	0.467320	0.46732	0.06447042
Run 9	0.4672338	0.4671832	0.06447044	0.467320	0.46732	0.06447042
Run 10	0.4674419	0.4672967	0.06447043	0.467320	0.46732	0.06447042

Fig. 13 and Fig. 14 show the mean fitness and best fitness with respect to iteration for CPSO and APSO respectively for Test Function 3.

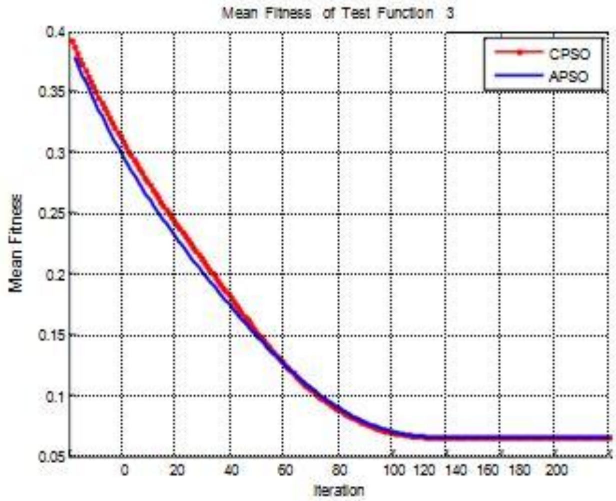

Fig. 13 Mean of different fitness at each iteration for CPSO and APSO for Test Function 3

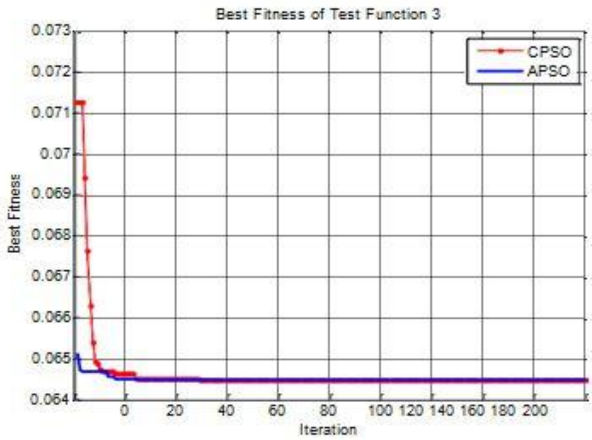

Fig. 14 Best fitness at each iteration for CPSO and APSO for Test Function 3

Test Function 4 (Griewank):

Fig. 15 shows the iteration of each algorithm to hit 1% tolerance band of best fitness value for 10 different runs.

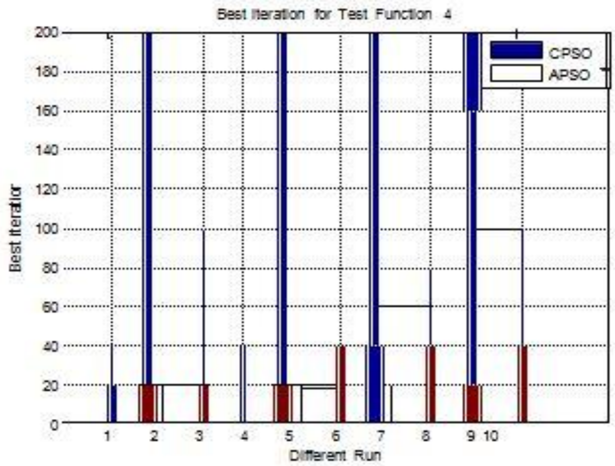

Fig. 15 Iteration to enter 1% tolerance band of global minima for CPSO and APSO for Test Function 4

Table 4 shows the position in two dimensional area for best fitness value or minima for test function 4 to find out by optimization algorithm and in this table best fitness value find out by CPSO and APSO are shown as well as corresponding positions of these.

Table- 4 Best fitness value and corresponding positions for CPSO and APSO for Test Function 4

TF 4	CPSO			APSO		
	Position		Best Fitness value	Position		Best Fitness value
	x_1	x_2	f(x)	x_1	x_2	f(x)
Run 1	-0.00136	-0.01001	0	0	-0.01029	0
Run 2	-3.10259	-4.36442	0.145669	0.00116	-0.01763	0
Run 3	0.007380	0.02453	0.000180	-0.04956	0.00618	0.00125
Run 4	-0.011823	-0.00949	0	-0.04630	0.02061	0.00119
Run 5	3.113180	4.35487	0.145622	0.00574	-0.01767	0
Run 6	-0.00906	0.00925	0	0	0	0
Run 7	-3.11065	-4.34979	0.1456275	0.00982	0.00040	0
Run 8	-0.00815	0.00561	0	0.014741	0.01827	0.00019
Run 9	3.113811	-4.35181	0.145628	-0.00718	0.03245	0.00029
Run 10	-0.00347	0.00610	0	-0.00288	-0.03167	0.00025

Fig. 16 and Fig. 17 show the mean fitness and best fitness with respect to iteration for CPSO and APSO respectively for Test Function 4.

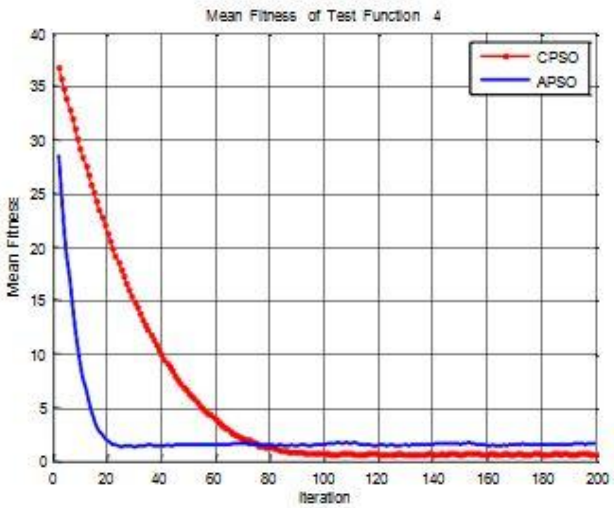

Fig. 16 Mean of different fitness at each iteration for CPSO and APSO for Test Function 4

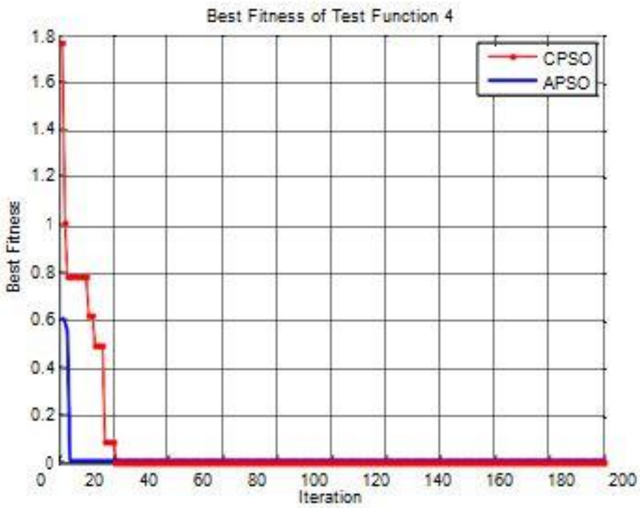

Fig. 17 Best fitness at each iteration for CPSO and APSO for Test Function 4

On the basis of all above results, it can be concluded that APSO gives better results in comparison of CPSO. There are 5 different Test functions considered of different shapes for examination the optimization algorithm. In these Test functions, proposed APSO optimization algorithm shows better results.

VI. CONCLUSION

Finding optimum performance is most prime requirement in of any system and in day to day increasing complexity of these systems made it as challenging task. Many optimization techniques are used for this purpose but PSO is shown it suitable for this purpose due to simplicity and fast convergence rate. In this work, CPSO and APSO are explored to find global optimum for different benchmark test functions. APSO shown better performance than CPSO in terms fast convergence rate and does not bound to local optima. There are 5 different well known benchmark test functions have been used for present study. Nature of these objective functions differs from each other. APSO showed better performance for these test function than it will show better performance for digital communication system also.

REFERENCES

[1] R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization, Swarm Intell., 2007, 1(1), 33–57.
[2] M. Dorigo, V. Maniezzo, and a Colorni, Ant system: optimization by a colony of cooperating agents., IEEE Trans. Syst. Man. Cybern. B. Cybern., 1996, 26(1), 29–41.
[3] R. C. Eberhart and Y. Shi, Particle swarm optimization: developments, applications and resources, in Proceedings of the 2001 congress on evolutionary computation, 2001, pp. 81–86.
[4] J. Kennedy and R. Eberhart, Particle swarm optimization, in IEEE International conference on Neural Networks, 1995, 4, 1942–1948.
[5] M. Clerc and J. Kennedy, The particle swarm - explosion, stability, and convergence in amultidimensional complex space, IEEE Trans. Evol. Comput., 2002, 6(1), 58–73.
[6] M. S. Arumugam and M. V. C. Rao, On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems, Appl. Soft Comput., 2008, 8(1), 324–336.
[7] M. Jamil and X. Y. Blekinge, A Literature Survey of Benchmark Functions For Global Optimization Problems, Int. J. Math. Model. Numer. Optim., 2013, 4(2),150–194.

