Advanced Particle Swarm Optimization Algorithm with improved velocity update strategy

Talha Ali Khan
School of Biomedical Engineering
University of Technology Sydney
Sydney, Australia
talha.khan@uts.edu.au

Sai Ho Ling
School of Biomedical Engineering
University of Technology Sydney
Sydney, Australia
steve.ling@uts.edu.au

Ananda Sanagavarapu Mohan
School of Biomedical Engineering
University of Technology Sydney
Sydney, Australia
Ananda.Sanagavarapu@uts.edu.au

***Abstract*—In this paper, Advanced Particle Swarm Optimization Algorithm (APSO) with improved velocity updated strategy is presented. The algorithm incorporates an improved term in the velocity update equation so that the particles will reach the optimum point quickly and convergence is much faster as compared with the Standard PSO (SPSO) and other improved PSOs. Five benchmark functions are given to evaluate the efficiency of the suggested algorithm. The results of simulation empirically demonstrated that the proposed technique has remarkably improved in terms of convergence rate and solution quality.**

***Keywords*—APSO, benchmark functions, swarm technique**

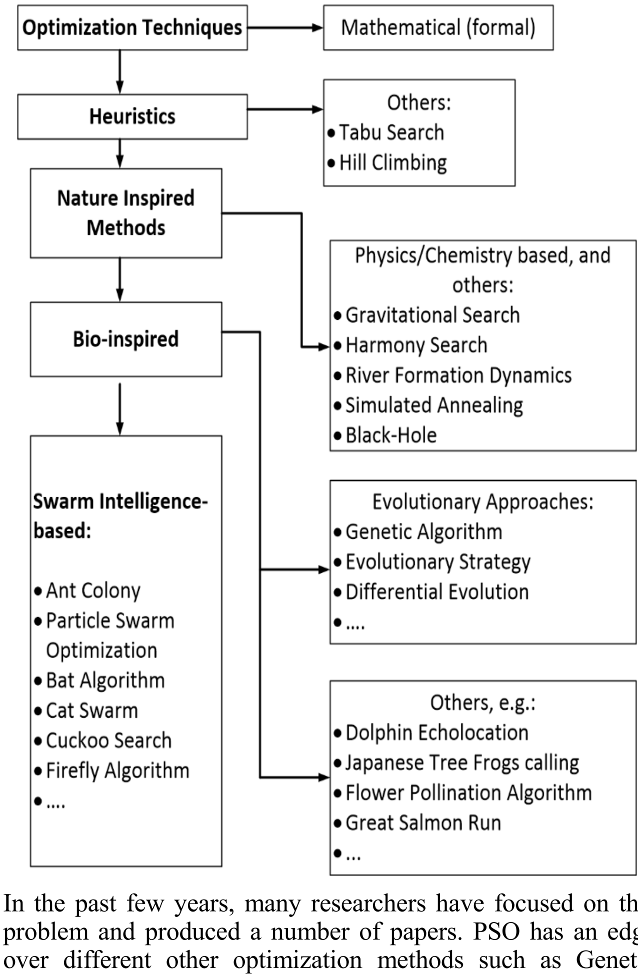
I. INTRODUCTION

Due to increasing complexity in the biomedical applications, optimization problems are getting famous in the real world issues. Continuous efforts are required in order to develop smart optimization techniques and therefore it is still an open research area. Figure 1 highlights a few approaches in optimization methods. The most significant types of the optimization approaches come directly from different observations and phenomenon of the nature and is known as Swarm Intelligence (SI). The word "swarm" originated due to the non-uniform motion of characterised by particles in the search capacity which is more resembling to the scourge of mosquitoes , group of fishes or flock of birds [1]. In this paper, the focus is on one of the swarm intelligence based approach that is acknowledged as Particle Swarm Optimization (PSO).

PSO is a progressive computational approach which is presented by Kennedy and Eberhart in 1995, the approach was incline towards the social behaviour of different animals . The roots of the PSO meets with social psychology, artificial life and for both computer science and engineering [2]. It structure of the PSO works on the idea of particles called "population" that move in the problem search space boundaries by given some velocities. After every iteration, each particle updated its velocity according to its previous best position and best position of the neighbourhood. As the particles move in the search space these movements bring the particles close to an approximately nearer or to an optimal solution [3].

PSO is an intelligent computational dependent approach which is not primarily influence by the nonlinear nature and size of the problem, and can easily be converged to an optimal point in most of the problems where other techniques fail to provide an optimal solution [4]. Hence, it can be used efficiently to various optimization tasks in engineering and other related fields.

Fig 1 shows different types of nature inspired optimization algorithms



In the past few years, many researchers have focused on this problem and produced a number of papers. PSO has an edge over different other optimization methods such as Genetic Algorithm. Following are the advantages of PSO:

a) It requires only a few parameters adjustment and therefore it is easy to implement.

b) It has a good memory capability as compared to GA since it stores every particle own previous best and neighbourhoods' best values.

c) Since all the particles utilize the information of the best particle to improve themselves by doing so they are keeping the idea of social communication in the community whereas in GA the bad solutions are thrown out and just best solutions are saved as a result, in GA the population revolves around the subset of the best individuals.

The particle velocity is a presumptive variable and due to this reason, it creates an uncurbed path allowing the members to make wider cycles in the search space. To avoid these oscillations the limits for the velocity is implemented in terms of upper and lower limits in the APSO. The other reason of modifying the standard equation of velocity and adding a new term is that during exploration the particles got stuck in the local minima and cannot escape. The concept of increasing velocity and helps the particles to reach the optimal position faster and improve the convergence.

II. ELEMENTARY CONCEPTIONS

PSO depends upon two basic disciplines that are computer science and social sciences. Moreover, it uses the swarm intelligence idea that is the characteristics of a system, whereas the mutual actions of the callow agents that are working together in the neighborhood with their environs produce clear global functional patterns [5]. Hence, the foundations of can be defined as under:

i) Societal Ideas: It is identified as with the social communications human intelligence is evolved. Assessment, judgement, and emulation of others, getting knowledge from surroundings and understanding of other behaviors permit humans to familiarize to the environs and find out optimum patterns of behavior and approaches. Additionally, another important societal perception show that the "beliefs and awareness are indivisible significances of human sociality." Beliefs and principles are engendered when persons become further alike due to common social culture. These permit people to move nearer to the adaptive patterns of behavior.

ii) Swarm Intelligence Foundation: SI can be defined by five basic ideas.

a) Quality Concept: the particles should be capable to reply to the quality aspects in the environs.

b) Constancy Concept: the particles would not alter its way of behavior at any time as the situation varries.

c) Concept of diversity: the particles should not obligate its activity along extremely narrow channels.

d) Flexibility Concept: the particles would be capable of varying its mode of behaviour whenever it is needed. Particles do not have any mass and volume (or having a very negiotiable mass or volume) and are subject to velocities and accelerations towards a better mode of behavior.

e) Proximity Concept: the particles should be capable to perform simple space and time computations.

iii) Computational Characteristics: SI gives a suitable pattern for implementing adaptive systems. It is an enhancement of evolutionary computation and contains the softening parameterization of logical operators like AND, NOT and OR. Particularly, PSO is an enhancement, and a possibly significant existence of cellular automata (CA). The particle swarm can be intellectualized as cells in CA, whose conditions transforms in many dimensions at the same time. Equally, PSO and CA share the subsequent computational characteristics [6].

a) Individual particles are updated in parallel.

b) The updated value of each particle depends upon the last values of particles and its neighbor.

c) Same rules are applied for all the updates.

III. STANDARD PARTICLE SWARM OPTIMIZATION ALGORITHM (SPSO)

In PSO the solution is modelled in the form of particles which can move throughout the search space. Particles position can be find out by its position vector and its movement by its velocity.

$$x_{id}(k+1) = x_{id}(k) + v_{id} \quad (1)$$

The experience of the each particle is based upon its own decision and the information of the other particles in its surroundings. Both these factors have equal significance and can be changed depending upon particles decision so it is practicable to use random numbers on each part, hence, the velocity equation will be

$$v_{id} = wv_{id} + c_1R_1(p_{id} - x_{id}) + c_2R_2(p_{gd} - x_{id}) \quad (2)$$

where R_1 , R_2 are two random numbers with uniform distribution in the range of [0-1], xi_d is current position of the particle , v_{id} is velocity of the particle, p_{id} is the local position of the particle , p_{gd} is the global best position of the particle, K is the iteration number, w is inertial weight and c_1, c_2 are two positive acceleration constants numbers.

Following is the flowchart to implement the PSO algorithm.

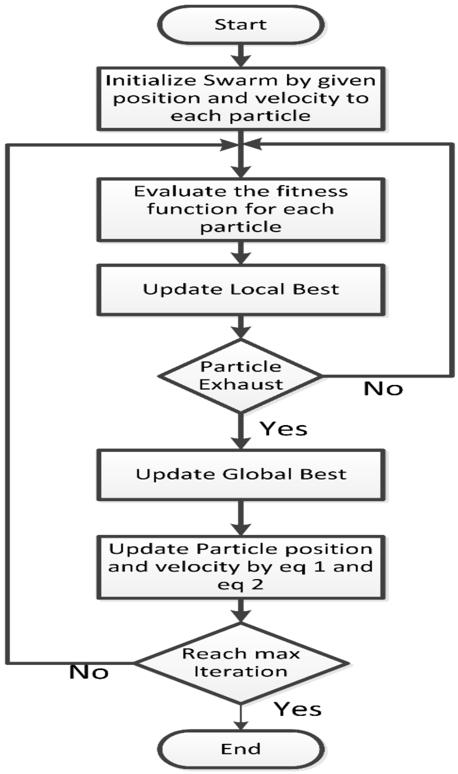


Fig 2 Flow Chart for Implementing PSO

The velocity of the particles can be update by equation 2 and it has three vital parts.

1) Inertia Part: This part is known as the habit or momentum. It helps the particle to travel in the same direction in which it has been moving. By a constant in the modified versions of PSO this part can be scaled.

2) Cognition Part: This part is known as remembrance, memory or self-knowledge. It helps the particle to find out its best position whose relative fitness value is also known as the particle's best position scaled by a random weight.

3) Social Part: This part is called as collaboration , social knowledge or mutual information. This part helps to find out best position by any particle whose relative fitness value is also known as the global best position scaled by a random weight.

IV. ADVANCED PARTICLE SWARM OPTIMIZATION ALGORITHM (APSO)

The standard PSO algorithm is primarily based on the two equations position and the velocity of the particle. In this paper, a term is added in the velocity equation, which improve the performance of the PSO.

a) Modification in the velocity update equation:

The initialization technique for the advanced PSO is the same as with standard PSO and using the same parameters. The primary difference is in the velocity equation that is also the key part of the PSO algorithm. The third term that is added in the velocity equation of the PSO is used to minimize the positions of the particles through the iterations so the velocity well be increased and the algorithm well be reach to the optimal solution faster. Moreover, the inertia weight is also used as a factor to the particle location. The velocity and position equations will be

$$v_{id} = wv_{id} + c_1R_1(p_{id} - x_{id}) + c_2R_2(p_{gd} - x_{id}) + w\left(\frac{c_1}{c_2}\right)(p_{id} - p_{gd}) \quad (3)$$

$$x_{id}(k+1) = wx_{id}(k) + v_{id} \quad (4)$$

The new method used different equations and approaches to determine the particle new position and velocity. As mentioned above initializing parameters, the local and global best will be find out in the same way by the fitness test as with the standard PSO. The reason for dividing C_1 and C_2 in the velocity updated equation in the modified term is to make the value neither very smaller nor very large since both of the acceleration constants have a huge impact on the particles' movement in the search space. A nominal value help the particles to converge faster and quicker towards the optimal solution.

b) Important Parameters selection for the proposed algorithm:

A number of the factors have taken into account while implementing the APSO algorithm. These factors include topology, inertia constant, acceleration constants and limiting maximum velocity.

i. Topology of the APSO:

There are two common kinds of the neighbourhoods in which the particles performance is noted: a) local best and 2) global best. In the global best approach, the best position is find by any of the particle and the rest of the population is attracted towards this position. That shows a wider network in which each member of the swarm can able to access the information by the other members of the swarm. On the other hand, in local best case each member has access to the information of its close neighbours, depending upon the category of the swarm topology [7]. The other two general topologies are the wheel topology in which the members are not communicating with each other but the correspondence is taken place to a focal individual, ring topology is another approach in which each member is connected with two other neighbours only [8]. For the APSO in this paper, Global best approach is implemented so the particles have broader access of the information of the entire members of the swarm. However, selecting the best effective topology for the members in swarm based on the type of the problem. One topology may work better in a given problem and perform deplorable in the others.

ii. Maximum velocity Selection:

After every iteration, the APSO algorithm works by estimating the velocity and position of each member of the swarm that they move in the given dimensions in the search space. If the value is very large than members of the swarm can travel unsteady and moving far away from the good solution; on the contrary if it is very small it limits the movement of the particle and they do not travel towards the best solution.

iii) Acceleration constants Selection:

Acceleration constants in the velocity equation manage the motion of every member in the swarm approaching the global and local best position. If the values of the acceleration constants is too small it will reduce the motion of the each member on the other hand if it is too larger then the each member diverge from its position. The acceleration constants are often set to be such that will satisfy $C_1+C_2 \leq 4$. If the constraints is not $C_1+C_2 \leq 4$ is not satisfied then PSO does not usually converge. That is why in APSO the values of the C_1 and C_2 are chosen as 2.1 and 1.9 respectively.

iv) Selection of inertia constant:

To observe convergence behaviour of the proposed APSO inertial weight "*w*" is very significant parameter. Inertial weight controls the impact of the previous velocity on the current update. Inertial weight is a trade-off between global and local abilities of the swarm. If the inertial weight has a larger value it will result in facilitating global exploration, i.e. it will help in searching new areas. Similarly, if the inertial weight has small value it will result in facilitating local exploration i.e. it will help in fine-tuning the current search area. Therefore, in APSO the inertial weight is linearly decreased from current iteration to next iteration. Two different parameters are being defined which are w_{max} and w_{min} .The following relation is being used for inertial weight control:

$$w = w_{max} - \left(\frac{w_{max}-w_{min}}{FE}\right) * i \quad (5)$$

In eq (5) [9], *i* is the current function evaluation. The value of w_{max} and w_{min} has been optimized to achieve the best results. Constriction factor has also been introduced in this paper as a modification made by [10]. The constriction factor is calculated as follows when phi=4.1;

$$chi = \frac{2}{phi-2+\sqrt{phi^2-4*phi}} \quad (6)$$

This constriction factor is used to modify inertial weight. Following control is incorporated in the inertial weight.

$$w = chi * \left(0.0005 + w * \left(\frac{FE-(i-30)}{FE}\right)\right) \quad (7)$$

Maximum number of function evaluation is used as the stopping criteria. In this paper, the value of phi is constant 4.1.

V. RESULTS AND ANALYSIS

To compare and evaluate the performance of the proposed algorithm with the standard and other improved version of PSO different benchmark functions are used.

a) Benchmark Functions

The consistency, effectiveness, and performance tests of various optimization techniques are outlined by using different benchmark functions and standards. In this way, the quality of the solution and the convergence are calculated. The test functions used for the performance evaluation of the proposed algorithm are presented below.

i) Sphere:

Sphere is a unimodal function that is symmetrical model with a single minimum.

$$f_1(x) = \sum_{i=1}^{D} x_i^2 \quad (8)$$

ii) Ackley:

Ackley is a multi-modal function with many local minima.

$$f_2(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e \quad (9)$$

iii) Rastrigin :

Rastrigin is also a multi-modal function with many local minima.

$$f_3(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10] \quad (10)$$

iv) Griewank

It is a multi-modal function with many local minima therefore; it is inclined to convergence in wrong direction.

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (11)$$

v) Rosenbrock:

It is a unimodal function with a single minimum.

$$f_5(x) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (12)$$

b) Simulation Setup

The performance of the proposed APSO, the Breed PSO [11], the Fuzzy PID PSO [11] , the Particle Swarm Optimization Cuckoo Search Paralleled Algorithm (PSOCSPA) [12] and the Cooperative Coevolving Particle Swarm Optimization (CCPSO2) [13] are evaluated on the five benchmark test functions.

The following simulation conditions are used:

- Number of Particles=30;
- Acceleration Constant $C_1 = 2.1$;
- Acceleration Constant $C_2 = 1.9$;
- Dimension D=30;
- Function Evaluation: 5000*D.

c) Statistical Analysis:

The results for the APSO and its comparison with standard and the other improved PSO algorithms will be presented. The results are compared in terms of standard deviation, mean and *t*-value as shown in in Table 1 and Table 2.

TABLE 1 shows the comparison between APSO, SPSO and other improved

Functions		APSO	SPSO	Breed PSO	Fuzzy PID PSO	CCPSO2	PSOCSPA
$f_1(x)$	Mean	$1.36e^{-159}$	0.0057	-	-	$3{,}27e^{-6}$	$1.33e^{+01}$
	Std. Dev	$4.30e^{-159}$	0.0178	-	-	$1{,}07e^{-5}$	$7.30e^{+01}$
$f_2(x)$	Mean	$6.21e^{-16}$	0.0209	14.440	10.00	$7{,}93e^{-1}$	$3.841e^{-06}$
	Std. Dev	$1.96e^{-15}$	0.629	13.526	0.0	$3{,}68e^{0}$	$5.964e^{-06}$
$f_3(x)$	Mean	4.814	$1.94e^{+2}$	-	-	$1{,}68e^{1}$	0.0
	Std. Dev	15.224	$1.08e^{+2}$	-	-	$9{,}08e^{1}$	0.0
$f_4(x)$	Mean	0.0090	$1.35e^{2}$	-	-	$5{,}36e^{-6}$	$9.696e^{-07}$
	Std. Dev	0.02846	$2.23e^{2}$	-	-	$1{,}97e^{-2}$	$3.060e^{-6}$
$f_5(x)$	Mean	2.770	$2.30e^{+3}$	-	-	$3{,}47e^{2}$	-
	Std. Dev	8.760	$2.12e^{+3}$	-	-	$6{,}65e^{2}$	-

PSO Algorithms.

The other testing criteria used is to calculate the *t*-test value. This method is used to evaluate the performance criteria between the two algorithms. By using the standard deviation and the mean values of the two algorithms, the t-value is calculated. If the t-value is negative between the two algorithms then the first algorithm has a poor credibility than the first algorithm and if is positive then the first algorithm is superior to the second algorithm. The terms used in calculating t-values are: ξ value of the degree of freedom, α_1 and α_2 are the mean values and, σ_1 and σ_2 are the values for the standard deviation of the two algorithms. The t-value can be expressed as:

$$t = \frac{\alpha_1 - \alpha_2}{\sqrt{\left(\frac{\sigma_1^2}{\xi + 1}\right) + \left(\frac{\sigma_2^2}{\xi + 1}\right)}}$$

If the t-value is larger than 1.645, which means that, the first algorithm performance is superior to the second algorithm by 95%. The t-values between the APSO, SPSO and various other versions of improved PSO methods are shown in Table II. It is noticeable that the t-value for the proposed algorithm is higher than 1.645 for most of the functions.

TABLE II T-Values between APSO and Other PSO Methods

Functions	t-value between APSO and SPSO	t-value between APSO and Breed PSO	t-value between APSO and Fuzzy PID PSO	t-value between APSO and CCPSO2	t-value between APSO and PSOCSPA
$f_1(x)$	0.2264	-	-	2.165	1.288
$f_2(x)$	2.3495	7.54	NA	1.523	4.55
$f_3(x)$	12.26	-	-	1.304	-2.23
$f_4(x)$	4.28	-	-	-0.22	-2.13
$f_5(x)$	7.23	-	-	3.689	-

d) Graphical Analysis

From figure 3 to figure 7 the comparison between the SPSO and APSO is presented. It is clear from the figures that the APSO gives better performance than SPSO and shows faster convergence for all the five tested benchmark functions.

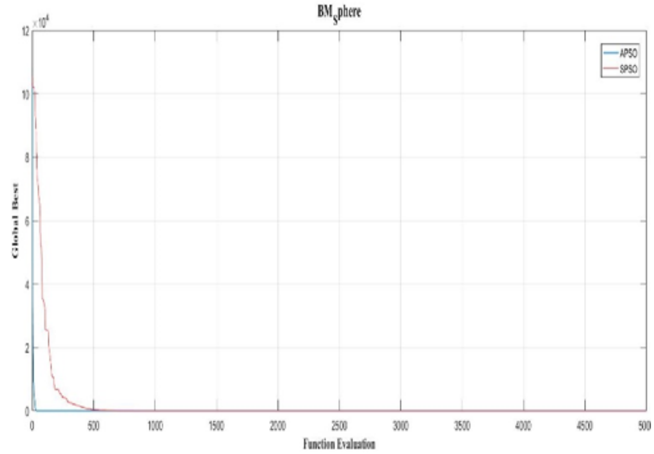


Figure 3 shows a comparison between APSO and SPSO for Sphere Function

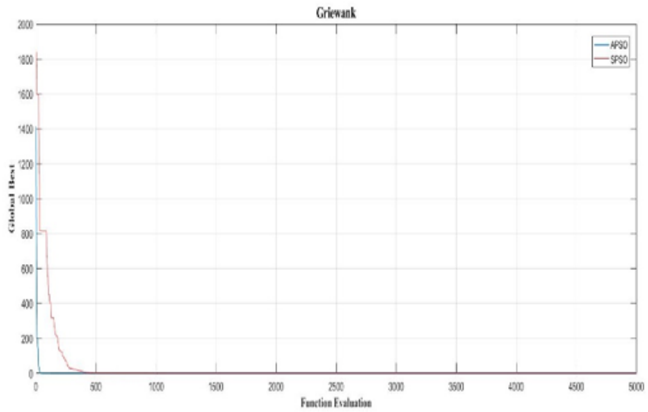


Figure 4 shows a comparison between APSO and SPSO for Griewank Function

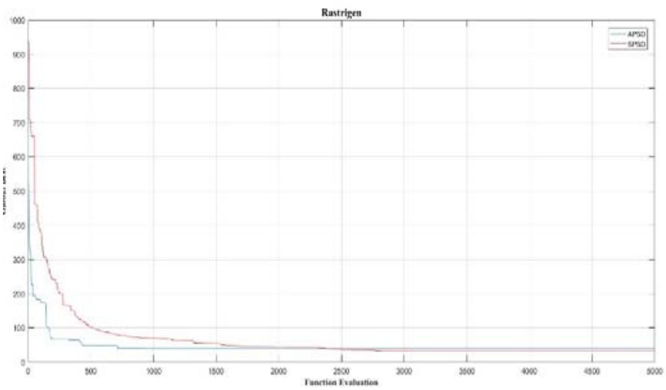


Figure 5 shows a comparison between APSO and SPSO for Rastrigin Function present excellent performance on the different types of problems.

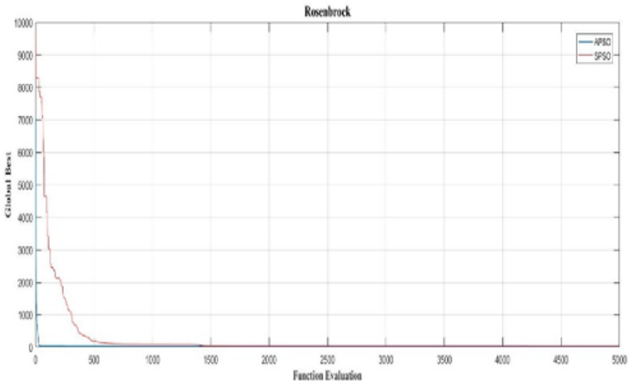


Figure 6 shows a comparison between APSO and SPSO for Rosenbrock Function

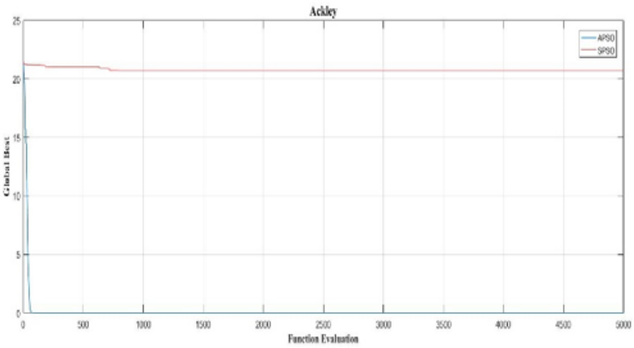


Figure 7 shows a comparison between APSO and SPSO for Rosenbrock Function

VI. CONCLUSION

PSO is an efficient tool for finding the optimized solution in a given problem hyperspace. A new term is added in the velocity update equation of the PSO in this paper. The purpose of adding this new term is to make the convergence of the PSO faster and make the solution quality better. Five benchmark functions are used to evaluate the performance of both standard and advanced PSO. It has been noticeable that the improved PSO performed better as compared to the standard and other improved version of the PSO in the literature in terms of different parameters such as t-values, mean and standard deviation. The advanced algorithm converges faster and effectively to the optimal solution. The advanced PSO also give better results almost on all the benchmark functions. They showed fast convergence, sphere being a unimodal function demonstrate rapid convergence even on higher dimensions. PSO is a prospective research topic and still has the capacity for new improvements and variations in the original algorithm that present excellent performance on the different types of problems.

VII. REFERENCES

[1] K. Ma, S. Liu, Y. Lin, Z. Yu, and K. Ji, "Parallel Grouping Particle Swarm Optimization with Stream Processing Paradigm," in *2017 IEEE 19th International Conference on High Performance Computing and Communications Workshops (HPCCWS)*, 2017, pp. 22-26.

[2] J. Guo and B. Wang, "Particle Swarm Optimization with Gaussian Disturbance," in *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, 2017, pp. 266-269.

[3] J. Wang, "Enhancing Particle Swarm Algorithm for Multimodal Optimization Problems," in *2017 International Conference on Computing Intelligence and Information System (CIIS)*, 2017, pp. 1-6.

[4] M. Sabeti, R. Boostani, and B. Davoodi, "Improved particle swarm optimisation to estimate bone age," *IET Image Processing,* vol. 12, no. 2, pp. 179-187, 2018.

[5] N. Bharanidharan and H. Rajaguru, "Performance analysis of particle swarm optimization technique in classification of dementia using MRI images," in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 311-315.

[6] L. Yuan, A. G. Miguel, P. Prem, S. Kazuo, and X. Yan, "Particle swarm optimization-based continuous cellular automaton for the simulation of deep reactive ion etching," *Journal of Micromechanics and Microengineering,* vol. 25, no. 5, p. 055023, 2015.

[7] E. M. N. Figueiredo and T. B. Ludermir, "Effect of the PSO Topologies on the Performance of the PSO-ELM," in *2012 Brazilian Symposium on Neural Networks*, 2012, pp. 178-183.

[8] L. Qingxue, B. J. v. Wyk, and S. Yanxia, "Small world network based dynamic topology for particle swarm optimization," in *2015 11th International Conference on Natural Computation (ICNC)*, 2015, pp. 289-294.

[9] M. Alhussein and S.H, "Improved Particle Swarm Optimization Based on Velocity Clamping and Particle Penalization," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, 2015, pp. 61-64.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69-73.

[11] Z. Wang, Q. Wang, D. He, Q. Liu, X. Zhu, and J. Guo, "An Improved Particle Swarm Optimization Algorithm Based on Fuzzy PID Control," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, 2017, pp. 835-839.

[12] Y. Xiaodong and C. Zefan, "Particle swarm optimization and cuckoo search paralleled algorithm," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 2236-2240.

[13] M. R. D. Ullmann, K. F. Pimentel, L. A. d. Melo, G. d. Cruz, and C. Vinhal, "Comparison of PSO variants applied to large scale optimization problems," in *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2017, pp. 1-6.