I. J. Mathematical Sciences and Computing, **2022, 2, 28-36**
Published Online June 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2022.02.03

Adaptive Social Acceleration Constant Based Particle Swarm Optimization

Jyoti Jain
Maharaja Surajmal Institute of Technology, Electrical and Electronics Engineering Department, Delhi, 110058, INDIA
E-mail: jyotieee@msit.in

Uma Nangia
Delhi Technological University/Electrical Engineering, Delhi, 110042, INDIA
E-mail: uma_nangia@rediffmail.com

N. K. Jain
Delhi Technological University/Electrical Engineering, Delhi, 110042, INDIA
E-mail: vnarender84@yahoo.com

Received: 25 November 2021; Revised: 17 December 2021; Accepted: 01 January 2022; Published: 08 June 2022

Abstract: In this paper, an attempt has been made to develop an Adaptive Social Acceleration Constant based PSO (ASACPSO). ASACPSO converge faster in comparison to basic PSO. The best value has been selected based on the minimum number of kounts required to minimize the function. Adaptive Social Acceleration Constant based PSO (ASACPSO) has been developed using the best value of adaptive social acceleration constant. The Adaptive Social Acceleration Constant has been searched using three formulations which led to the development of three algorithms-ALDPSO, AELDPSO-I and AELDPSO-II. All three were implemented on Rosenbrock function to get the best value of adaptive social acceleration constant. Similarly it has been implemented on seven mathematical benchmark functions and its performance has been compared to Basic Particle Swarm Optimization (BPSO). ASACPSO was observed to converge faster and give better accuracy. Results show that Kounts required for convergence of mathematical function is lesser for ASACPSO in comparison to basic PSO.ASACPSO reduces the computational time to optimize the function.

Index Terms: Convergence, Inertia weight, Optimization, Population, Particles, Social acceleration constant.

1. Introduction

The concept of Particle Swarm Optimization (PSO) was introduced by [1] Eberhart and Kennedy. PSO is based on five basic principles- proximity, quality, diversity, stability, and adaptability. An overview of particle swarm optimization was suggested by [2] Riccardo poli *et* al. References [3,4] suggested reviews on PSO. Reference [5] suggested application of PSO in the Engineering and Computer science. Reference [6] represents the studies of the variants of PSO like constant inertia weight, linear inertia reduction, limit on maximum velocity, constriction factor, dynamic inertia and maximum velocity reduction. The authors observed that constriction and dynamic inertia weight both affect reliability and cost. Reference [7] proposed modification of PSO using simultaneous perturbation optimization algorithm. The technique proposed had good global search and effective local search capability. [8] Zhi-Xiang – hou developed an adaptive Particle Swarm Optimization algorithm and claimed this to be more effective and highly accurate. [9] Md. Sakhawat Hossen *et al*. also suggested an adaptive Particle Swarm Optimization based on behaviour of spider. They presented a comparison with traditional PSO and suggested methods to improve the performance of adaptive PSO. In references [10,11] N.K.Jain *et al*. has developed PSO based on initial selection of particles and applied on mathematical benchmark functions and on Economic load dispatch problem. Reference [12] N.K.Jain *et al*. has suggested application of PSO for Multi-objective Economic Load dispatch problem. Reference [13] shows the application of PSO to solve MELD Problem. Reference [14] shows the application of ASACPSO to solve Economic Load Dispatch Problem. Reference [15] shows the Impacts of PSO Parameters on its Convergence. Reference [16] suggested improved model of PSO to give steady convergence and a better . Reference [17] shows the Self-Adaptive Acceleration Constants based PSO. In this both the constants have been adaptive and giving execution time slightly higher than others technique.

In this paper, an Adaptive Social Acceleration Constant based PSO (ASACPSO) has been developed. The best value of social acceleration constant has been searched and selected based on the minimum number of kounts required to minimize the function. Adaptive Social Acceleration Constant based PSO (ASACPSO) has been developed using the best value of adaptive social acceleration constant.Three formulations have been used to search the best adaptive social acceleration constant which is selected based on minimum number of kounts required to minimize the function. Kount is the number of function evaluations, i.e. how many times a function has been evaluated. To check the performance of ASACPSO has been tested on seven mathematical benchmark functions and compared with basic PSO (BPSO). ASACPSO has been found to perform better in terms of convergence and accuracy.

1.1 Basic Particle Swarm Optimization (Bpso)

BPSO is a population based self-adaptive, stochastic optimization technique. Its basic idea is mathematical modelling and simulation of food searching activities of a swarm of birds (particles). In the multidimensional space where the optimal solution is sought, each particle in the swarm is moved towards the optimal point. At each iteration, the particle moves towards the optimum solution, through its personal best solution obtained by itself and global best solution obtained by all the particles until they find food (optimum solution). In an n-dimensional search space, position and velocity of particle j is represented by vectors X_j=(X_{j1} , X_{j2},…… X_{jn}.) and V_{j1}=(V_{j1} V_{j2} ,… V_{jn}) respectively. Let X_{pbest} and X_{gbest} be the personal and global best position of particle j. The modified velocity and position of each particle can be calculated using current velocity and distance from X_{pbest} and X_{gbest} as follows:

$$V_{ij}^{k+1}=W * V_{ij}^{k}+C_p r_p(X_{pbestij}^{k}-X_{ij}^{k})+ C_g r_g(X_{gbestij}^{k}-X_{ij}^{k}) \quad i=1,2,\ldots..N \quad j=1,2\ldots..p \tag{1}$$

Velocities are updated by eq. (1) and position of each particle is updated using (2)

$$X_{ij}^{k+1}=X_{ij}^{k}+V_{ij}^{k+1} \tag{2}$$
$$i=1, 2\ldots.N; \quad j=1,2\ldots..p$$

Where
W: Inertia Weight.
K:teration Count
V_{ij}^{k+1} Velocity of j^{th} particle of i^{th} variable at k^{th} iteration.
X_{ij}^{k} Position of j^{th} particle of i^{th} variable at k^{th} iteration.
C_p: Cognitive acceleration coefficients.
$X_{pbestij}^{k}$: Personal best position of j^{th} particle of i^{th}variable in k^{th} iteration.
X_{gbesti}^{k} : Global best position of i^{th} variable until k^{th} iteration.
N: Total number of variables.
p : Number of particles in swarm.s
r_p, r_g: Two separately generated uniformly distributed random numbers.
F: Function value.

2. Development of Adaptive Social Acceleration Constant Based Pso (Asacpso)

In this paper, an adaptive social acceleration constant based PSO (ASACPSO) has been developed. Cg regulates the maximum step size in the direction of the global best particle. In basic PSO, C_p and C_g is fixed to 1.0, whereas in Adaptive Social Acceleration Constant based PSO (ASACPSO), Cg is searched adaptively and is named as adaptive social acceleration constant Csg. The adaptive social acceleration based constant Csg has been formulated using following three functions:

1. Linearly decreasing function

$$Csg1= cgmax-(i*(Cgmax-Cgmin) / it) \tag{3}$$

2. Exponential of linearly decreasing function

$$Csg2 = exp((-1)* (Cgmax-(i*(Cgmax-Cgmin)) / it)) \tag{4}$$

3. Exponential decreasing function consisting of exponential of constant multiplied by linearly decreasing function.

$$Csg3 = exp((-0.34)*(cgmax-(i*(cgmax-cgmin)/it))); \tag{5}$$

Where,
IT: Maximum number of iterations
I: Current iteration
Cgmax: Maximum value of Social acceleration constant
Cgmin: Minimum value of Social acceleration constant
Csg: Adaptive social acceleration constant
Csg1: Social acceleration constant based on linearly decreasing function
Csg2: Social acceleration constant based on exponential of linearly decreasing function
Csg3: Social acceleration constant based on exponentially decreasing function consisting of exponential of constant multiplied by linearly decreasing function.

2.1 Methodology

PSOs based on Csg1, Csg2, and Csg3 have been named as Adaptive Linearly decreasing PSO (ALDPSO), Adaptive Exponential Linearly Decreasing PSO (AELDPSO-I) and Exponential decreasing function consisting of constant multiplied by linearly decreasing function is called as Adaptive Exponential linearly decreasing PSO -II (AELDPSO-II) respectively.

All the three algorithms have been implemented to minimize the Rosenbrock function. The best value of Adaptive–Social acceleration constant Csg is searched by applying equation (3), (4) and (5) after second, third, fourth and upto seventh iteration. When these equations are applied after seventh iteration, there was no improvement in the results. The results of ALDPSO, AELDPSO- I, and AELDPSO- II are shown in Table 1, 2 and 3 respectively. The best value of Csg is selected based on minimum number of kounts required to minimize the function. Adaptive Social Acceleration Constant based PSO (ASACPSO) is then developed using the best value of Csg.

Kount is number of function evaluations, i.e. how many times a function has been evaluated. This is taken as the measure of computational time. The measure of computational time in seconds or minutes will depend on the technology, i.e. the type of computer being used. Therefore, the time has been measured in terms of function evaluations.

All the three algorithms are using following fixed value of parameters as follows:
Cgmax=1
Cgmin=0
IT=Maximum number of Iterations=1000
Population Size=40
Results for Mathematical Rosenbrock function using ALDPSO are shown in Table 1.

Table 1. Results Of Rosenbrock Function Using Aldpso

Condition for implementation of Csg1 (1)	**Results (2)**	**Itr (3)**	**Kount (4)**	**Csg1 (5)**
Iter>2	Sucess	45	1840	0.955
Iter>3	Sucess	46	1880	0.954
Iter>4	Sucess	42	1720	0.958
Iter>5	Sucess	**40**	**1640**	**0.96**
Iter>6	Sucess	48	1960	0.952
Iter>7	Sucess	44	1800	0.956

Column (1) of Table 1 shows the number of iterations after which Csg1 (defined by equation (3)) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Column (3) and (4) represent number of iterations and number of kounts required respectively to optimize the function . Column (5) shows value of Csg1 searched for Rosenbrock function. When linear decreasing function Csg1 implemented after second iteration kounts required to optimize the function is 1840 and an adaptive social acceleration constant value is 0.955 as shown in row(2) of Table 1.

By observing column (4) of Table 1 minimum kount is obtained when adaptive Csg has been searched after five iterations and its value is 0.96. It has been highlighted in the Table 1. When this strategy is applied after 8^{th} iteration, no improvement has been observed in results.

Results for Mathematical Rosenbrock function using AELDPSO-I is shown in Table 2.

Table 2. Results of Rosenbrock Function Using Aeldpso-I

Condition for implementation of Csg2 (1)	Results (2)	Itr (3)	Kount (4)	Csg2 (5)
Iter>2	Failure	1000	40040	0
Iter>3	Sucess	68	2760	0.3428
Iter>4	Sucess	54	2200	0.348
Iter>5	Sucess	62	2520	0.345
Iter>6	Sucess	64	2600	0.3443
Iter>7	Sucess	**44**	**1800**	**0.956**

Column (1) of Table 2 shows the number of iterations after which Csg2 (defined by equation (4)) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Column (3) and (4) represents number of iterations and number of kounts required respectively to optimize the function. Column (5) shows value of Csg2 searched for Rosenbrock function. By observing column (4) of Table-2 minimum kount is obtained when adaptive Csg2 is searched after seven 7^{th} iterations and its value is 0.956.The row corresponding to this has been highlighted. When this strategy is applied after 8^{th} iteration, no improvement has been observed in results.

Results for Mathematical Rosenbrock function using AELDPSO-II are shown in Table-3.

Table 3. Results of Rosenbrock Function For Aeldpso-Ii

Condition for implementation of Csg3 (1)	Result (2)	Itr (3)	Kount (4)	Csg3 (5)
iter>2	Sucess	33	1360	0.688
iter>3	Sucess	33	1360	0.79
iter>4	Sucess	36	1480	0.688
iter>5	Sucess	**30**	**1240**	**0.788**
iter>6	Sucess	32	1320	0.793
iter>7	Sucess	40	1640	0.7154

Column (1) of Table 3 shows the number of iterations after which Csg3 (defined by equation (5)) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Column (3) and (4) represent number of iterations and number of kounts required respectively to optimize the function. Column (5) shows value of Csg3 searched for Rosenbrock function. When Csg3 implemented after five iteration kounts required optimizing the function is 1240 and an adaptive social acceleration constant value is 0.788 as shown in row (4) of Table 3. By observing column (4) of Table 3 minimum kount is obtained when adaptive Csg has been searched after five iterations and its value is 0.788. It has been highlighted in the Table 3. When this strategy is applied after 8^{th} iteration, no improvement has been observed in results.

Comparison of results of ALDPSO, AELDPSO-I and AELDPSO-II are shown in Table 4.

Table 4. Comparison of Results of Aldpso, Aeldpso-I And Aeldpso-Ii

	ALDPSO		AELDPSO-I		AELDPSO-II	
Condition for implementation of Csg (1)	Csg1 (2)	Kount (3)	Csg2 (4)	Kount (5)	Csg3 (6)	Kount (7)
iter>2	0.955	1840	0	40040	0.688	1360
iter>3	0.954	1880	0.3428	2760	0.79	1360
iter>4	0.958	1720	0.348	2200	0.688	1480
iter>5	**0.96**	**1640**	**0.345**	**2520**	**0.788**	**1240**
iter>6	0.952	1960	0.3443	2600	0.793	1320
iter>7	0.956	1800	0.956	1800	0.7154	1640

A column (1) of Table 4 represents the number of iterations after which Csg is implemented. Column (2), (4) and (6) represent the adaptive social constants Csg1, Csg2 and Csg3 achieved by ALDPSO, AELDPSO-I, and AELDPSO-II respectively for each condition of implementation shown in column 1. Column (3), (5) and (7) represent the kounts required to minimize the function. It is observed that AELDPSO-II requires least kounts as compared to ALDPSO and AELDPSO-I for all conditions of column (1). It is further observed from Table 4 that for case of Iter>5, minimum kounts are required by AELDPSO-II. For this condition of Csg implementation, the kounts required by three algorithms are :

ALDPSO: 1640
AELDPSO-I: 2520
AELDPSO-II: 1240

This row has been highlighted. The value of an adaptive social acceleration constant is achieved by AELDPSO-II and is taken as social acceleration constant for PSO which leads to the development of ASACPSO.

3 Results and Discussions on Implementation of Basic Pso, Aeldpso-Ii and Asacpso on Different Mathematical Benchmark Functions:

The Experimentation has been done on following seven mathematical benchmark functions using Basic PSO, AELDPSO-II and ASACPSO.

1. Rosenbrock function
2. Beale Function
3. Booth Function
4. Hyper Ellipsoid Function
5. Ackley function
6. Schwefel function
7. Three Hump Function

Following parameters are taken for basic PSO (BPSO).

1. $r_p = C_g = r_g = C_p = 1$
2. W=0.6
3. Population size=40

These parameters have been selected merely for comparison of the results obtained by the proposed algorithms. Initially Basic PSO (BPSO) has been applied for 40 particles. Then the best social acceleration constant was searched using AELDPSO-II. Then this best value of social constant worked as Csg for ASACPSO. All the three algorithms BPSO, AELDPSO-II and ASACPSO have been implemented on seven mathematical benchmark functions and their results have been compared in Table 5 to 11. For all types of PSOs Cognitive acceleration coefficient is kept fixed i.e. Cp=1.

1. Rosenbrock function:

Mathematically, it is defined as

$$f=100*(x_1^2 - x_2)^2+(1 - x_1)^2$$

f (1,1)=0 Range: $x_1 \geq 0, x_2 \leq 1$

Table 5. Results for Rosenbrock Function

PSOs (1)	x1 (2)	x2 (3)	f (4)	Cg/Csg (5)	Kount (6)	Itr (7)
BPSO	0.99	0.99	1.9e-9	1	1600	39
AELDPSO-II	0.99	0.99	5.9e-10	0.788	1240	30
ASACPSO	0.99	0.99	1.79e-8	0.788	1320	32

Column (1) of Table 5 shows the type of PSO being implemented on Rosenbrock function. Column (2) and (3) represents two variables of Rosenbrock function. Column (4) represents the corresponding function value. Column (5) represents Cg for BPSO and Csg for AELDPSO-II and ASACPSO. Column (6) and (7) represents Kounts and iteration required respectively to optimize the function.

2. Beale Function:

Mathematically Beale function defined as

$$f=(1.5 - 1 + x_1x_2)^2+(2.25 - x_1 + x_1x_2^2)^2+\ldots(2.625 - x_1 - x_1x_2^3)^2$$

f (3,0.5)=0. Range $x_1 \geq -4.5$, $x_2 \leq 4.5$

Table 6. Results For Beale Function

PSOs (1)	x1 (2)	x2 (3)	F (4)	Cg/Csg (5)	Kount (6)	Itr (7)
BPSO	2.99	0.5	5.4e-10	1	1800	44
AELDPSO-II	3	0.5	9.42e-10	0.8071	1440	35
ASACPSO	3	0.49	1.1e-8	0.8071	1000	24

3. Booth Function:

Mathematically, it is defined as

$$f=(x_1 + 2x_2 - 7)^2+(2x_1 + x_2 - 5)^2$$

f(1,3)=0 Range x1$\geq$ -10, x_2<10

Table 7. Results for Booth Function

PSOs (1)	x1 (2)	x2 (3)	F (4)	Cg/Csg (5)	Kount (6)	Itr (7)
BPSO	1	3	2.06e-13	1	2000	49
AELDPSO-II	1	2.99	5.9e-10	0.7358	1320	32
ASACPSO	0.99	2.99	6.32e-8	0.7962	1280	31

4. Axis Parallel Hyper Ellipsoid Function:

Mathematically, it is defined as

$$f=\sum_{i=1}^{n} i\, x_i^2$$

f(x)=0 at i =1: n here n=2 Range $-5.2 \leq x_i \leq 5.12$

Table 8. Results for Axis Parallel Hyper Ellipsoid Function

PSOs (1)	x1 (2)	x2 (3)	f (4)	Cg/Csg (5)	Kount (6)	Itr (7)
BPSO	-1.165e-5	-7.33e-9	3.8e-11	1	1880	46
AELDPSO-II	-2.5e-6	-6.12e-6	8.15e-12	0.669	1680	41
ASACPSO	6.3e-5	6.3e-9	1.8e-9	0.669	1160	28

5. Ackley Function:

Mathematically, it is defined as

f(x1, x2)= -20exp(-0.2 √(0.5(x1^2)+x2^2))

f(0,0)=0 Range $-5.0 \leq x_i \leq 5.0$

Table 9. Results for Ackley Function

PSOs (1)	x1 (2)	x2 (3)	f (4)	Cg/Csg (5)	Kount (6)	Itr (7)
BPSO	-1.64e-11	-8.15e-12	5.204e-11	1	3920	97
AELDPSO-II	-1.40e-9	1.286e-9	5.3902e-9	0.725	2360	58
ASACPSO	6.09e-6	-2.692e-6	1.88e-5	0.725	2240	55

6. Schwefel Function:

Mathematically, it is defined as

f(x)=418.9829-x1*sin($\sqrt{abs}$(x1))

f(x)=0; at x=(420.9687)

Range $-500.0 \leq x_i \leq 500.0$

Table 10. Results for Schwefel Function

PSOs (1)	x1 (2)	x2 (3)	F (4)	Cg/Csg (5)	Kount (6)
BPSO	420.968	1.222e-5	1	3600	89
AELDPSO-II	420.965	1.4392e-5	0.7242	40	51
ASACPSO	6.09e-6	1.27e-5	0.7242	40	53

7. Three Hump Camel Function:

Mathematically, it is defined as

$$f(x1,x2)=2x1^2-1.05x1^4+x1^6/6+x1*x2+x2^2$$
$$f(0,0)=0$$

Table 11. Results for Three Hump Camel Function

PSOs (1)	x1 (2)	x2 (3)	F (4)	Cg/Csg (5)	Kount (6)	PSOs (1)
BPSO	-1.8062e-6	2.20e-6	7.40e-12	1	1800	44
AELDPSO-II	-2.236e-6	1.42e-5	1.82e-10	0.719	1320	32
ASACPSO	-2.1779e-6	-4.07e-7	1.27e-5	0.719	1000	24

Comparison of kounts for all the functions optimized by BPSO, ALDPSO-II and ASACPSO are shown in Table 12

Table 12. Comparison of Kounts For Mathematical Functions

S.No. (1)	Functions (2)	BPSO (Kounts) (3)	AELDPSO-II (Kounts) (4)	ASACPSO (Kounts) (5)
1	Rosenbrock	1600	1240	1320
2	Beale	1800	1440	1000
3	Booth	2000	1320	1280
4	Axis Parallel hyper ellipsoid	1880	1680	1160
5	Ackley Function	3920	2360	2240
6	Schwefel function	3600	40	40
7	Three Hump Camel function	1800	1320	1000

It is observed that minimum kounts are required for ASACPSO except in case of Rosenbrock function in which case kounts required for AELDPSO-II are minimum.

Table 13 shows the % saving in kounts obtained using ASACPSO and compared to BPSO

Table 13. Results of Bpso and Asacpso And Saving In % Of Kount

S. No. (1)	Functions (2)	BPSO (3)	ASACPSO (4)	Saving in % of Kount
1	Rosenbrock	1600	1320	17.5
2	Beale	1800	1000	44.5
3	Booth	2000	1280	36
4	Axis Parallel hyper ellipsoid	1880	1160	38.29
5	Ackley	3920	2240	42.85
6	Schwefel	3600	40	97.8
7	Three Hump Camel function	1800	1000	44.5

It has been observed that every function converged for lesser iterations and lesser kounts for ASACPSO in comparison to basic PSO (BPSO). Minimum saving in kount for Rosenbrock function and maximum saving in kount for Schwefel function has been found as shown in Table 13.

A saving of 44.5% has been achieved in Beale function. The saving has been calculated by following formula. Saving in kount %=[100*(Kounts for 40 particles (with cg=1) – kounts for 40 particles with Csg1)] / kounts for 40 particles(with cg=1).

Comparison of kounts for different functions for BPSO & ASACPSO are shown in Fig.1.

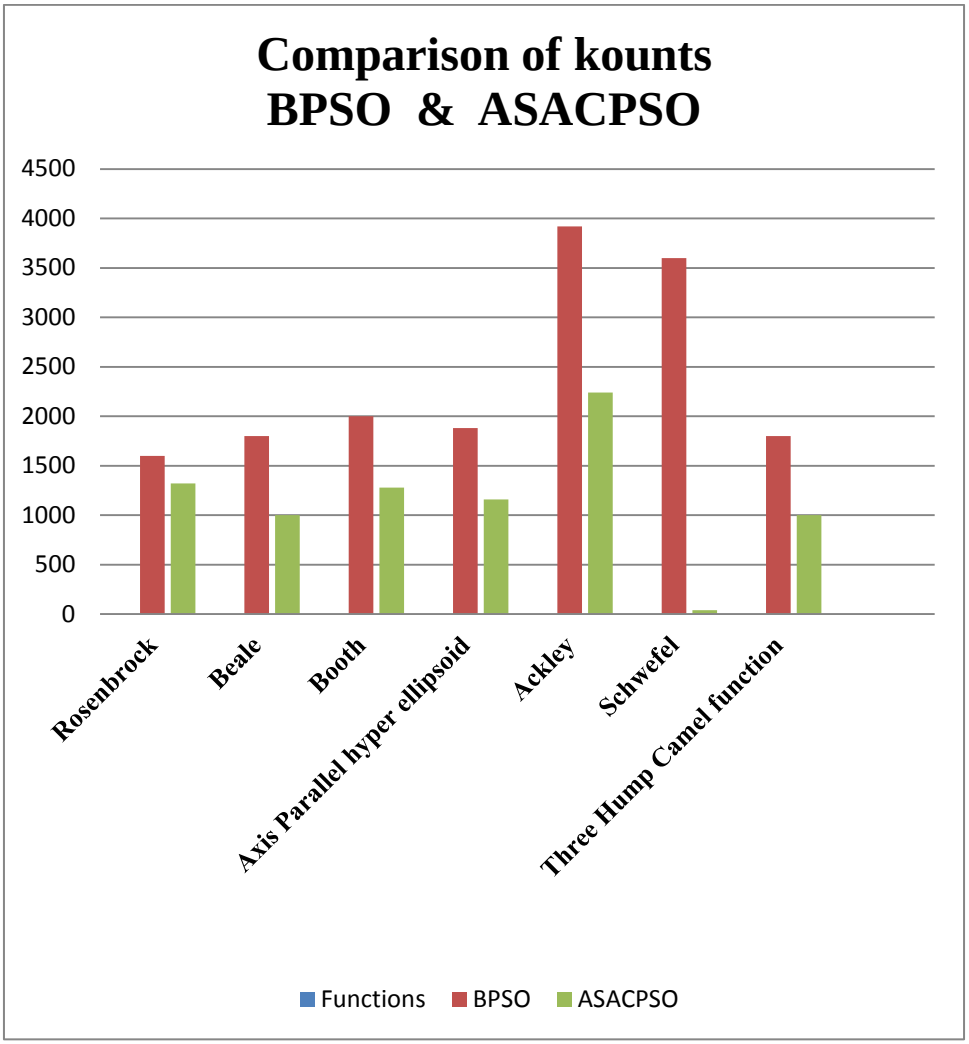

Fig. 1. Comparison of kounts for different functions for BPSO & ASACPSO.

4. Conclusions

1. In this paper ALDPSO, AELDPSO-I, AELDPSO-II have been developed and were applied to Rosenbrock Function. It was observed from the results of Rosenbrock function that AELDPSO-II minimized the function in less kount in comparison to ALDPSO and AELDPSO-I.
2. ASACPSO was developed using best value of Social Acceleration Constant obtained by AELDPSO-II .
3. ASACPSO was applied to the seven mathematical benchmark functions.
4. It was observed that AELDPSO-II and ASACPSO converge faster in comparison to basic PSO.
5. It has also been observed that saving in kounts is obtained for all the functions. Maximum saving of 97.6% in kounts is obtained for Schwefel function. However, for Rosenbrock function minimum saving of 17.5% kounts is achieved.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization", Proc. IEEE International Conference on Neural networks (Perth, Australia), 1995, IEEE Service Center, Piscataway, NJ, pp. 1942-1948.
[2] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization. An overview," Swarm Intelligence, vol. 1, no. 1, pp. 33–57, 2007.
[3] Dian Palupi Rini, Siti Mariyam Shamsuddin, and Siti Sophiyati Yuhaniz, "Particle Swarm Optimization: Technique System and Challenges", International Journal of Computer Applications (0975-8887) Vol-14 – January 1, 2011.
[4] Jain N.K., Nangia Uma, Jyoti Jain, "A Review of Particle swarm optimization", Journal of The Institution of Engineers (India): Series B, August, 2018, 99(4), pp.407-411. ISSN 2250-2106, DOI 10.1007/s40031-018-0323-y.
[5] Russell C. Eberhart, and yuhui shi, "Particle Swarm Optimization: Developments, Applicationsand Resources"Evolutionary Computation,2001 Proceeding of the 2001 Congress on IEEE,1(2001),pp.81-86

[6] Jaco F. Schutteand Albert A.Groenwold, "A Study of Global Optimization Using Particle Swarms", Journal of Global Optimization, Vol. 31, Springer 2005, pp. 93-108.
[7] yitaka Maeda and Naoto Matsushita , "Empirical Study of Simultaneous Perturbation Particle Swarm Optimization", SICE Annual Conference, August 20-22, 2008, pp. 2545-2548.
[8] Zhi-Xiang Hou, "Wiener model identification based on adaptive particle swarm optimization", Proceedings of the Seventh IEEE International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008, pp. 1041-1045.
[9] Md. Sakhawat Hossen, Fazle Rabbi, and Md. Mainur Rahman, "Adaptive Particle Swarm Optimization(APSO) for multimodal function optimization", International Journal of Engineering and Technology, Vol.1, No. 3,2009, pp. 98-103.
[10] N.K.Jain, Uma Nangia, Jyoti Jain "An improved PSO based on initial selection of particles (ISBPSO) for Economic Load Dispatch" Proceedings of 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES 2016),4-6 july ,2016.
[11] N.K.Jain, Uma Nangia, Jyoti Jain, "An improved PSO based on initial selection of particles (ISBPSO)", Proceedings of 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES 2016),4-6 july ,2016.
[12] N.K.Jain, Uma Nangia, Ashwary jain "PSO for Multiobjective Economic Load Dispatch(MELD) for minimizing Generation Cost and Transmission Losses", journal of the Institution of Engineers (India):series B,2015.
[13] Jain N.K., Nangia Uma, Jyoti Jain, "Multiobjective Economic Load Dispatch Studies in 2-D and 3-D Space by Particle Swarm Optimization Technique", February, 2019, Journal of The Institution of Engineers (India): Springer Series B, DOI : 10.1007/s40031-019-00386-z.
[14] Jain N.K., Nangia Uma, Jyoti Jain, " Economic load Dispatch using Adaptive Social Acceleration Constant based Particle Swarm Optimization", Journal Of Institution Of Engineers (India): Series B, October 2018, 99(5), 431-439, ISSN 2250-2106, DOI 10.1007/s40031-018-0322-z.
[15] N. K. Jain, Uma Nangia, Jyoti Jain, "Impacts of PSO Parameters on its Convergence" IEEE second International Conference on Power Electronics, Intelligent Control and Energy Sysytems (ICPEICES-2018), Delhi, pp.1-5, (22-24 October, 2018).
[16] Shengli Song, Li Kong and Jingjing Cheng, A Novel Particle Swarm Optimization Algorithm Model with Centroid and its Application, I.J. Intelligent Systems and Applications, 2009, 1, 42-49 October 2009 in MECS(http://www.mecs-press.org/)
[17] Sudip Mandal, " A Modified Particle Swarm Optimization Algorithm based on Self-Adaptive Acceleration Constants" I.J. Modern Education and Computer Science, 2017, 8, 49-56 Published Online August 2017 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijmecs.2017.08.07

Authors' Profiles

Dr. Jyoti Jain, is from Delhi. She has completed Ph.D. from Delhi Technological University Delhi. She is working in the field of solution of Multiobjective and Single objective optimization problems using Intelligent Techniques. She has 28 years teaching experience. She is working as an Associate professor at Maharaja Surajmal Institute of Technology (MSIT) Delhi.

She is a senior member of IEEE and Chapter Advisor of IEEE PES Student branch Chapter MSIT. She has received outstanding Engineer Award from IEEE PES IAS Delhi Chapter in 2021.

Dr. Uma Nangia and Dr. N.K. Jain are working as a Professor at Electrical Engineering Department of Delhi Technological University. They are working in the field of Optimization of Economic Load Dispatch, Reactive Power and Power Systems.

How to cite this paper: Jyoti Jain, Uma Nangia, N. K. Jain," Adaptive Social Acceleration Constant Based Particle Swarm Optimization ", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.8, No.2, pp. 28-36, 2022. DOI: 10.5815/ijmsc.2022.02.03