International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Variants of Particle Swarm Optimization and Onus of Acceleration Coefficients

Y. V. R. Naga Pawan, Kolla Bhanu Prakash

Abstract**: *The Particle Swarm Optimization (PSO) is a widely used optimization algorithm for finding optimized solutions in a diverse gamut of problem domains. The parameters like Initialization, Constriction factor, Inertia Weight, Mutation Operator, Fuzzy Logic and Parallelism have engendered the Particle Swarm Optimization (PSO) with many variants. The variants of PSO have outperformed the Basic Particle Swarm Optimization. In order to comprehend the role of acceleration coefficients in BPSO, an inquiry is carried out. It is observed that the convergence speed of the BPSO is quicker when the acceleration coefficients are not equal than when both are equal.***

Index Terms**: *Acceleration Coefficients, Inertia Weight, Particle Swarm Optimization, Variants.***

I. INTRODUCTION

Across the globe, Reynolds [1], Boid Model and Kennedy and Eberhart [2] nature inspired metaheuristic algorithm, Particle Swarm Optimization (PSO), fascinated many researchers due its minimalism and vowing optimization fitness it is implemented in various domains and applications.

PSO simulates the behaviour of flock of birds, insects, herds etc. flying within the dimensional area in search of food or nest during a conformed cooperative manner. The learning experiences gained by the participant of the swarm and its neighbours, the participant changes its search pattern. The flocking model consists bird like objects called Boids [1]. Each Boid is defined by position and velocity. Boids have total awareness of events in some local vicinity. The Boids exhibit three simple steering behaviours, specifically, Separation, Alignment and Cohesion shown in Fig. 1, Fig. 2 and Fig. 3[1]. The Separation behaviour direct the flock mates to avoid crowding. The mechanism followed here at each time step, the boids will steer away from anyone in their individual space to avoid collision. The Alignment behaviour navigate the flock mates towards the average heading of local flock mates. The mechanism of behaviour at each time step, the boids will correct their alignment by some factor towards average alignment of the boids in their velocity. The Cohesion behaviour shove the flock mates towards the average position of local flock mates. The mechanism followed here at each time step, a tiny velocity is added towards 'Center of gravity' of the birds' local vicinity. The neighbourhood is characterized by distance and an angle. The flock mates outside the vicinity are ignored.

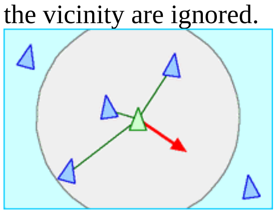

Fig. 1 Separation - Steer to prevent crowding participants of the local flock

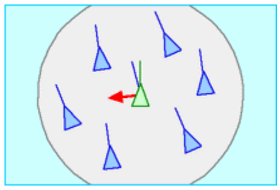

Fig. 2: Alignment – Steer towards the local flock mates' median direction

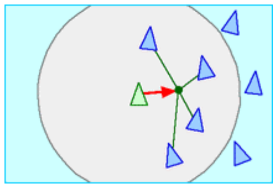

Fig. 3. Cohesion – Steer towards the local flock mates' median place

A. Basic PSO Algorithm

The PSO algorithm components are particle, Fitness function, local best, global best, velocity change and position change.

Each participant is called a particle and the populace is called as swarm, in PSO algorithm. With a random solution, each particle in the swarm is initialized. It moves through the problem space by adjusting its velocity dynamically with historical behaviours of itself and its neighbours. Mathematically, the PSO can be represented as follows:
Suppose N is the swarm size. Each particle position vector in D-dimensional space is represented as $PX_i = (px_{i1}, px_{i2}, px_{i3}, \ldots, px_{iD})$. The velocity vector is represented as $PV_i = (pv_{i1}, pv_{i2}, pv_{i3}, \ldots, pv_{iD})$. Each particle reminiscences its preceding best position of the j^{th} particle, $P_i = (p_{j1}, p_{j2}, p_{j3}, .., p_{jD})$ along each dimension. The particle best is known through 'pbest', the previous best value. The Swarms best value is known through 'gbest', the global best value. The following are the equations for updating the particle velocities and position:

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Y. V. R. Naga Pawan, Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vijayawada, INDIA.
Kolla Bhanu Prakash, Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vijayawada, INDIA.

$$pv_{id} = pv_{id} + c_1 * random() * (pbest_i - px_{id}) + c_2 * Random() * gbest_i - px_{id} \quad (1)$$

$$px_{id} = px_{id} + pv_{id} \quad (2)$$

where c1 and c2 are two acceleration coefficients, random() and Random() are two random functions in the [0,1]. pv_i is the clamped to a maximum velocity v_{max}, the parameter given by the user. In the equation (1), preceding velocity is the first portion, the cognition part of the particle is the second portion, and the cooperation among the particles as third portion [3]. In the Fig. 4, the flowchart for the basic PSO algorithm is shown, The flowchart has three sections, the first section is local aspect of the particle, second sections tells about particle vicinity and the last section tells about on global outlook of the swarm. The pseudocode of the BPSO is shown in the Fig. 5.

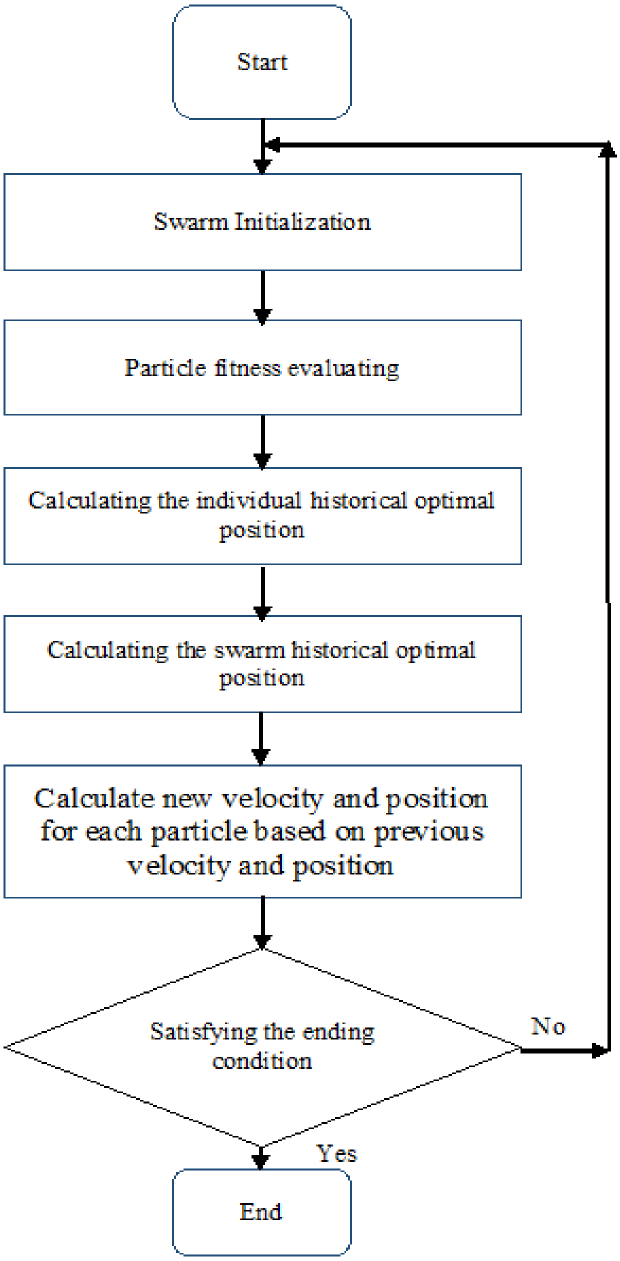

Fig. 4. Flowchart of Basic PSO

The flowchart has three parts, the first part is local, second part is based on vicinity and the last part on global. The pseudocode for basic PSO is given below:

Step 1:
Initialization
For each particle, i, in the population
Initialize X[i] with uniform distribution
Initialize V[i] randomly.
Evaluate the objective function of X[i] and assigned the value to fitness[i].
Initialize p_{pbest}[i] with a copy of X[i].
Initialize pbest_fitness[i] with a copy of fitness[i].
Initialize p_{gbest} with index of the particle with the least fitness.
Step 2:
Repeat until stopping criterion is reached
For each particle, i,:
Update V[i] and X[i] according to the equations (1) and (2)
Evaluate fitness[i]
If fitness[i] < pbest_fitness[i] then
Pbest[i] = X[i]
Pbest_fitness [i] = fitness[i]
Update pgbest by the particle with current least fitness among the population

Fig. 5: Pseudocode of Basic PSO

II. BENEFITS AND TRADE-OFFS OF PSO

PSO uses less resources than other optimization techniques and it do not guarantee optimal solution. The variants of PSO ensures the availability of global optimum, faster convergence and accuracy.

III. APPLICATIONS OF PSO

Poli[4] and Zhang et al [5] presented a birds eye view of applications on PSO. Table 1 presents various application areas of PSO in which literature is available.

Table 1 Applications of PSO

S.No.	Application Areas
1	Antennas
2	Automated Control Systems
3	Biological Engineering
4	Biomedical
5	Chemical Engineering
6	Clustering and Classification
7	Combinatorial optimization
8	Communication Networks
9	Design
10	Distribution Networks
11	Electrical Engineering
12	Electronics
13	Entertainment Optimization
14	Fault Optimization
15	Financial Analysis and Optimization
16	Fuel and Energy
17	Fuzzy and Neuro Fuzzy
18	Graphics and Visualization
19	Image and Video
20	Mechanical Engineering
21	Metallurgy
22	Modelling
23	Neural Networks
24	Power Systems and Plants

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication (BEIESP)

25	Prediction and Forecasting
26	Robotics
27	Scheduling
28	Security and Military
29	Sensor Networks
30	Signal Processing

IV. VARIANTS OF PSO

The parameters considered for variants of PSO considered are Initialization, Constriction Factor, and Inertia weight, Mutation Operators, Fuzzy Logic and Parallelism. The studied information is tabulated in Table 2 – Table 20.

V. BENCHMARK FUNCTIONS

The performance of PSO and its variants are evaluated based on various benchmark test functions. Their characteristics are based on converge rate, accuracy etc. The classification is made based on modality, basins, valleys, separability, and dimensionality [60].

Table 2 PSO Variants based on Parameter Initialization

Author	Method / Approach	Findings
Nguyen et al [5]	• Low Discrepancy Sequence Initialization • Used Sobol, Halton and Faure Sequences	• Sobol sequence found best among all the Techniques [5].
Jabeen et al [6]	• Particle swarm optimization based on opposition method (O-PSO)	• O-PSO outperformed with BPSO. • O-PSO converged faster.
Pant et al [7]	• Quasi-random or Low Discrepancy Sequences • Used Van der Corput (VC-PSO), Sobol (SO-PSO)	• VC-PSO and SO-PSO outperformed the BPSO. • In 60% of cases VC-PSO is superior to SO-PSO whereas in 40% cases SO-PSO is superior to VC_PSO.
Omran [8]	• Used Opposition based Learning • Proposed OPSO, iOPSO, iPSO.	• iOPSO and iPSO outperformed approaches. • Without using any additional parameters the performance of PSO is enhanced by using opposition based Learning.
Chang et al [9]	• Uses Quasi-Oppositional Comprehensive Learning PSO (QCLPSO)	• QCLPSO yields outstanding performance on multimodal problems with less susceptible to premature convergence
Diez at el [10]	• Deterministic Particle Swarm Optimization (DPSO)	• DPSO is effectiveness on large scale problems.
Moaath Shatnawi et al [11]	• Polar PSO	• Little enhancements is noticed and no variance by using Analysis of Variance (ANOVA).
Parsopoulos et al [12]	• Nonlinear Simplex Method (NSM)	• NSM guides the swarm optimizer faster. • Helps is detecting global minimizer even if PSO fails to find.
Shahzad et al [13]	• OVCPSO proposed • Uses velocity Clamping • Uses Opposition based learning	• OVCPSO efficiently and effectively dealt with unimodal and multimodal optimization problems. • In OVCPSO, the significant number of function calls (NFC) are lower than standard PSO with Inertia weight and velocity clamping, OPSO with Cauchy Mutation
Alhussein et al [14]	• Clamping the velocity of the particle. • Particle velocity is penalized if the summation of the vector velocity a nd position vector extends beyond the search space limits.	• The modified PSO is very fast in convergence with Inertia-weight constant or linearly decreasing.
Tang et al [15]	• Proposed an Enhanced Opposition PSO (EOPSO)	• EOPSO performs better than BPSO and OPSO.
Silva et al [16]	• Predator – prey PSO (PPPSO)	• PPPSO have better results over multimodal functions but not with unimodal functions.
Xu et al [17]	• Extended PSO (EPSO)	• EPSO evolves in a moderate pace.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 3 PSO Variants based on Constriction Factor

Author	Method / Approach	Findings
Clerc et al [18]	• An approach to balance exploration and exploitation by new parameter χ, Constriction factor to the BPSO.	• able to find the minima of some extremely complex benchmark functions
Raha et al[19]	• Applied Constriction factor to increase the velocity of the particle.	• Constriction factor is effective over BPSO.
Lim et al [20]	• CFBPSO • IPSO	• CFBPSO is minimising the cost of generation. • CFBPSO improves the convergence and performs better when compared with IPSO.
Mauro et al [21]	• Type 1 and Type 1" COPSO	• Type 1 and Type 1" COPSO improves convergence of solution faster,

Table 4 PSO Variants based on Inertia Weight

Author	Method / Approach	Findings
Yuhui Shi [22]	• To control exploration and Exploitation	• Converges to Optimal solution fastly for certain inertia weights.
H-R Li [23]	• Proposed PSO based on linearly decreasing inertia weight (LDIW PSO), exponent decreasing inertia weight (EDW-PSO) and exponent decreasing inertia weight & stochastic mutation (EDM-PSO) [23]	• EDM PSO comes out quickly from local optimum • EDM PSO convergences faster than LDIW PSO and EDW PSO.
Chongpeng et al [24]	• Decreasing Inertia Weight (DIW)	• Best results to balance global and local search.
Yuhui Shi [25]	• Fuzzy Adaptive PSO • Linearly Decreasing Inertia Weight (LDIW)	• Fuzzy system tuning its inertia weight improves basic PSO.
Zhang et al [26]	• Processing in Parallel • Random Number Inertia Weight (RNW).	• RNW based PSO has better results than LDIW. • RNW overcomes LDIW that local Search lacks. • RNW overcomed early stage execution and global search ability at the end of the execution using LDIW.
Wei et al [27]	• Dynamically changing Inertia Weight, DPSO • Local Optimum Dimension mutation operator is designed to escape from Local Optimum	• DPSO is faster and better than LDW. • The proposed dimension mutation operator is helped from escaping local optimum.
Pant et al [28]	• *GWPSO+ED, GWPSO+GD and GWPSO+UD.*	• *GWPSO+ED, GWPSO+GD and GWPSO+UD gave superior performance over BPSO.*
Xuedan Liu et al [29]	• Multistart PSO • Improved PSO with linearly decreasing Inertia	• Avoids early convergence and augments global search ability • The Performance of the PSO is of better-quality using LDIW from 0.9 to 0.4
Yadmellat et al [30]	• Fuzzy tuned Inertia Weight PSO (FIPSO)	• The performance of the BPSO is increased with a dynamical adjustable fuzzy strategy based Inertia Weight according as per the current iteration and mean relative velocity • FIPSO performed remarkably well over BPSO in terms of consistency and global optimum search.

Table 5 PSO Variant based on Niche Mechanism

Author	Method / Approach	Findings
Liao et al[31]	• Niching PSO • Promotes cross-trap capability	• Niching PSO is efficient and effective.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 6 PSO Variant based on Mutation Operator

Author	Method / Approach	Findings
Wang et al [32]	• Hybrid PSO (HPSO) with Cauchy mutation operator	• Introducing Cauchy Mutation Operator on the best particle of the PSO the mutated best particle could contribute to all the remaining particles to the better positions. • HPSO finds better solution than PSO.
Wang et al [33]	• OPSO using dynamic Cauchy mutation	• OPSO convergence more quickly on simple unimodal functions. • OPSO has better global search capability on multi-modal functions.
Pant et al [34]	• AMPSO1-mutates the personal best position of the swarm. • AMPSO2 – mutates global best position in the swarm. • Both algorithms uses adaptive mutation using beta distribution	• AMPSO2 gives better results for unconstrained test cases. • The distribution of beta probability is in line with the distribution of Gaussian and Cauchy.
Pant et al [35]	• SM-PSO • Systematic Mutation(SM) operator is used. • SM uses Quasi-random Sobol Sequence. • SMPSO1 and SMPSO2	• SMPSO1 and SMPSO2 better than BPSO and QPSO.
Wu et al [36]	• Uses Power Mutation • PMPSO is proposed	• PMPSO outpaces PSO and PSO with Cauchy mutation. • PMPSO demonstrates excellent efficiency on unimodal functions. • Multimodal functions cannot be solved by PMPSO.
Imran et al [37]	• Uses Opposition based initialization • Cauchy mutation operator • Power Mutation Operator	• In case of unimodal functions Power Mutation operator is better whereas for multi-modal functions the Cauchy mutation operator is good.
Imran et al [38]	• Student T Mutation • STPSO is proposed	• STPSO is significantly well over BPSO, CPSO (PSO with Cauchy Mutation), AMPSO
Chen [39]	• NMPSO is proposed • Uses Novel Mutation operator	• NMPSO achieves more accurate results than BPSO. • NMPSO converges faster.

Table 7 PSO Variant based on Fuzzy Logic

Author	Method / Approach	Findings
Liu et al [40]	• FPSO is proposed • Uses Fuzzy Inertia Weight Control and Fuzzy location updated control	• FPSO avoids local optima and converges with better search capability and accuracy.
Wang et al [41]	• FUZZY_PSO is proposed	• FUZZY_PSO solves group particle convergence and conflicts of diversity which prevented premature phenomenon. • FUZZY PSO is more rapidly evolving and more accurately converging.
Ji et al [42]	• BPSO-CC proposed	• BPSO-CC is better performed when compared to SPSO in convergence speed and precision.
Dashora et al [43]	• Adaptive PSO (APSO)	• APSO outperforms the BPSO.
Olivas et al [44]	• Type-2 Fuzzy Logic PSO (TFL PSO)	• TFL PSO outperforms standard PSO
Kumar et al [45]	• Fuzzy PSO (FPSO)	• FPSO outperforms BPSO in convergence rate, optima value and consistency

Table 8 PSO Variant based on Parameter

Author	Method / Approach	Findings
Yang et al [46]	• Introduction of parameters p and q in PSO	• If p<0.5 then PSO is appropriate for multimodal problems and if p>0.5 then PSO is appropriate for unimodal problems. • While changing particle velocity, q is raised to place distinct emphasis on social or personal aspect. Adaptive q is used for different applications.

Table 9 PSO Variant based on Multi Swarm

Author	Method / Approach	Findings
Liang et al [47]	• DMS-PSO	• Effectiveness and efficiency is obtained.

Table 10 PSO Variant based on Parallel Processing

Author	Method / Approach	Findings
Dazhi et al [48]	• Multi-population PSO (MPSO) • Adaptive Serial PSO • Adaptive Parallel PSO	• Adaptive Parallel PSO is better than Adaptive Serial PSO. • Adaptive Parallel PSO save more time than Adaptive Serial PSO.
Bhaskar et al [49]	• CONPSO Proposed	• CONPSO outperforms PSO and FDR-PSO
Fukuyama et al [50]	• Parallel PSO is proposed	• Fast computation
Ping et al [51]	• Bulk Synchronous Parallel Chaos PSO (BSPCPSO)	• BSPCPSO improves optimization capacity of the particles.
Singh et al [52]	• Parallel PSO (PPSO)	• PPSO is working.

Table 11 PSO Variant based on Topologies

Author	Method / Approach	Findings
Bastos-Filho et al [53]	• PSO-ELM • Global, Local, Von-Neumann, Four Cluster topologies are discussed	• Global Topology has good capacity of exploitation.
McNabb et al [54]	• Hearsay PSO	• Hearsay PSO enables particles to interact more than just their individual best, helping to distribute data even faster. • Topologies to be connected more densely, decreasing PSO's cost communication in a large scale concurrent environments.
Lin et al [55]	• HSPPSO	• HSPPSO is better than BPSO.
Janson et al [56]	• H-PSO is proposed	• H-PSO is promising.
Ren et al [57]	• PSO-BP is proposed. • Improved PSO-BP proposed	• Improved PSO-BP has higher accuracy and quicker response than traditional PSO-BP.

Table 12 PSO Variant based on Ensemble

Author	Variant Factor	Method / Approach	Findings
Alam et al [58]	Parallel and RLC Operator	• Rotate Left and Complement – RLC • Flip operator is proposed	• Avoids unnecessary computations. • Produced comparable and competitive result
Tian [59]	Fuzzy Logic and Inertia Weight	• Novel Fuzzy PSO (NFPSO)	• NFPSO is feasible and effective.
Vazquez et al [60]	Topology and Inertia Weight	• Global best (gbest) PSO • Local best (lbest) PSO • Topologies – Star, Ring, Von Neumann, Random	• The global best algorithm (gbest) implements the topology of the Star and the local best algorithm (lbest) implements the topologies of Von Neumann and Random.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 13 Convergence Time in seconds for f1 (sphere) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000610	0.000839	0.001068	0.000992	0.001144	0.001755	0.001678	0.002670
25	0.000916	0.001068	0.002823	0.002441	0.003357	0.004425	0.004654	0.006561
50	0.001678	0.001907	0.004654	0.005646	0.008164	0.009613	0.008850	0.014267
75	0.002899	0.003052	0.007477	0.009308	0.011597	0.015946	0.014954	0.023499
100	0.004578	0.004501	0.011521	0.013352	0.019455	0.023041	0.021592	0.034333

Table 14 Convergence Time in seconds for f2 (Step 3) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000000	0.000076	0.000153	0.000229	0.000458	0.000534	0.000610	0.000839
25	0.000076	0.000076	0.000610	0.000458	0.000916	0.001144	0.001450	0.001984
50	0.000153	0.000076	0.001144	0.000992	0.001907	0.002213	0.002747	0.003967
75	0.000153	0.000229	0.001602	0.001373	0.003510	0.003433	0.004425	0.006180
100	0.000381	0.000305	0.001678	0.001755	0.004578	0.004959	0.006485	0.008927

Table 15 Convergence Time in seconds for f3 (Step 2) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000076	0.000076	0.000229	0.000229	0.000458	0.000534	0.000534	0.000839
25	0.000000	0.000000	0.000381	0.000534	0.001602	0.001144	0.001297	0.001984
50	0.000305	0.000153	0.001297	0.000839	0.001907	0.002213	0.002747	0.004044
75	0.000229	0.000153	0.001755	0.001297	0.003052	0.003586	0.004578	0.005951
100	0.000381	0.000229	0.001831	0.001755	0.004578	0.004959	0.006409	0.009232

Table 16 Convergence Time in seconds for f4 (Dejong F4) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000153	0.000153	0.000458	0.000687	0.001144	0.001144	0.001297	0.001831
25	0.000458	0.000381	0.001144	0.001373	0.002289	0.002747	0.002823	0.004501
50	0.000839	0.000610	0.003815	0.003052	0.004578	0.005875	0.006027	0.009461
75	0.000763	0.000839	0.004654	0.004807	0.007095	0.010071	0.009842	0.015335
100	0.001144	0.001144	0.006867	0.007095	0.010910	0.013886	0.014649	0.022660

Table 17 Convergence Time in seconds for f5 (Alpine) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000839	0.001068	0.005112	0.008698	0.066529	0.017243	0.635157	0.017853
25	0.001907	0.002289	0.007248	0.010910	0.014954	0.068589	0.014267	1.648432
50	0.003967	0.004807	0.014801	0.021592	0.030976	0.054780	0.035019	0.105821
75	0.007935	0.007782	0.021973	0.032120	0.050278	0.081559	0.059434	0.123064
100	0.011292	0.010834	0.033799	0.043565	0.072175	0.115358	0.095674	0.192187

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Table 18 Convergence Time in seconds for f6 benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000000	0.000076	0.002747	0.003510	0.003815	0.005417	0.007248	0.008164
25	0.000000	0.000076	0.008469	0.009918	0.011063	0.014878	0.012131	0.018235
50	0.000153	0.000076	0.015335	0.021744	0.024567	0.034867	0.025864	0.044785
75	0.000229	0.000229	0.029679	0.033341	0.040665	0.060502	0.041581	0.069657
100	0.000229	0.000229	0.038376	0.046693	0.074846	0.088731	0.057832	0.099031

Table 19 Convergence Time in seconds for f7 (Quartic) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000229	0.000229	0.000534	0.000763	0.000916	0.001373	0.001373	0.002060
25	0.000458	0.000458	0.001602	0.001678	0.002594	0.003281	0.003128	0.005035
50	0.000610	0.000687	0.003204	0.003510	0.005035	0.006409	0.006867	0.011139
75	0.000992	0.000916	0.004807	0.005341	0.008087	0.010758	0.010910	0.016556
100	0.001602	0.001221	0.007019	0.007706	0.011597	0.014954	0.016251	0.023728

Table 20 Convergence time in seconds for f8 (Schwefel 2.26) benchmark function

Swarm Size	Dimension							
	1		2		3		4	
	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2	c1.NE.c2	c1.EQ.c2
10	0.000687	0.000916	0.000916	0.000916	0.002976	0.003586	0.004730	0.004578
25	0.002365	0.002289	0.001373	0.000610	0.007324	0.009613	0.006638	0.006485
50	0.004578	0.004807	0.001984	0.000610	0.016098	0.024643	0.011063	0.008774
75	0.006485	0.007706	0.001297	0.000763	0.027161	0.038224	0.016022	0.010224
100	0.008316	0.010681	0.001526	0.000839	0.035019	0.049210	0.018463	0.012818

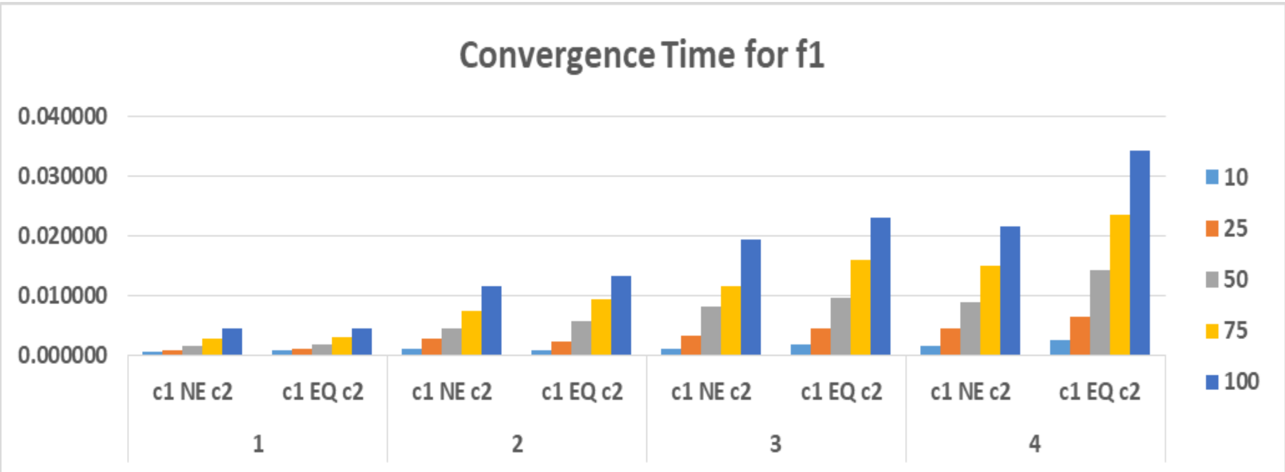

Fig. 6: Convergence of f1 (sphere) benchmark function in seconds

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

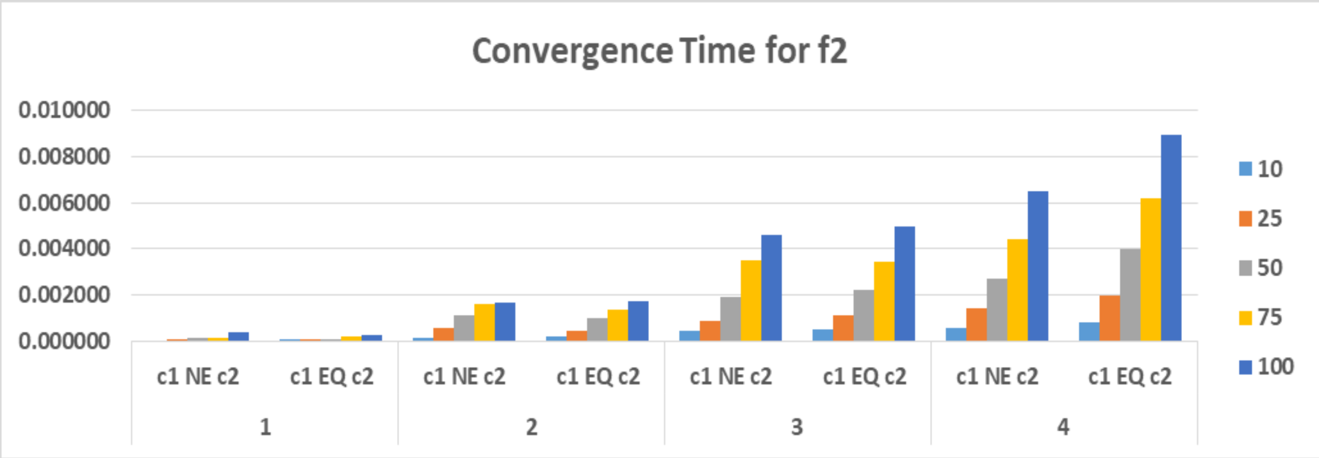

Fig. 7: Convergence of f2 (Step 3) benchmark function in seconds

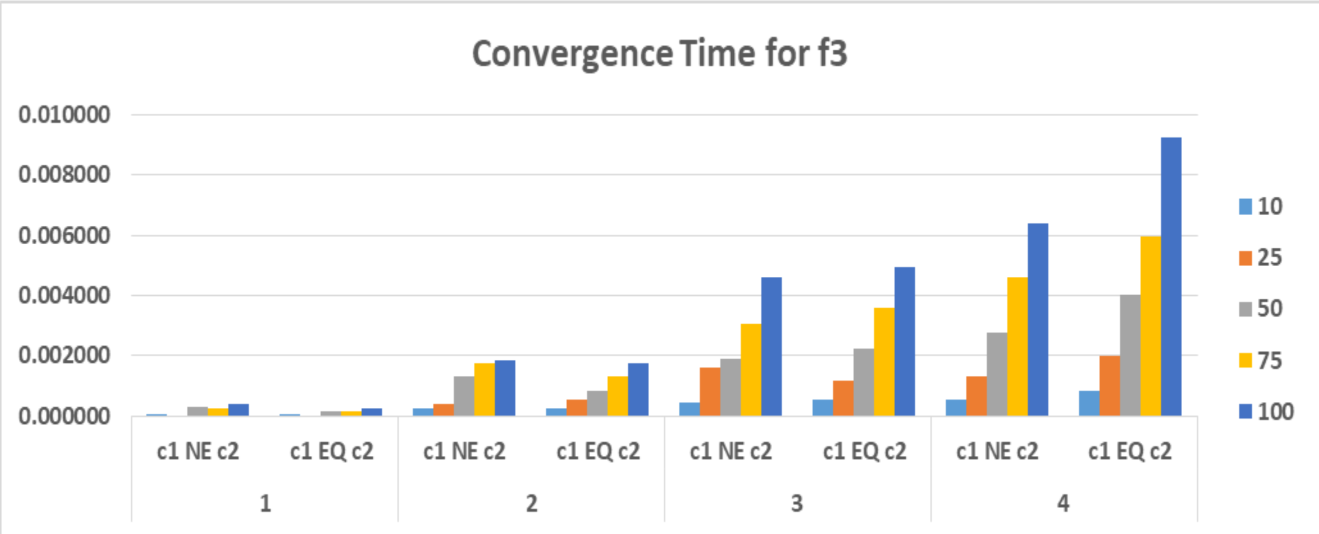

Fig. 8. Convergence of f3 (Step 2) benchmark function in seconds

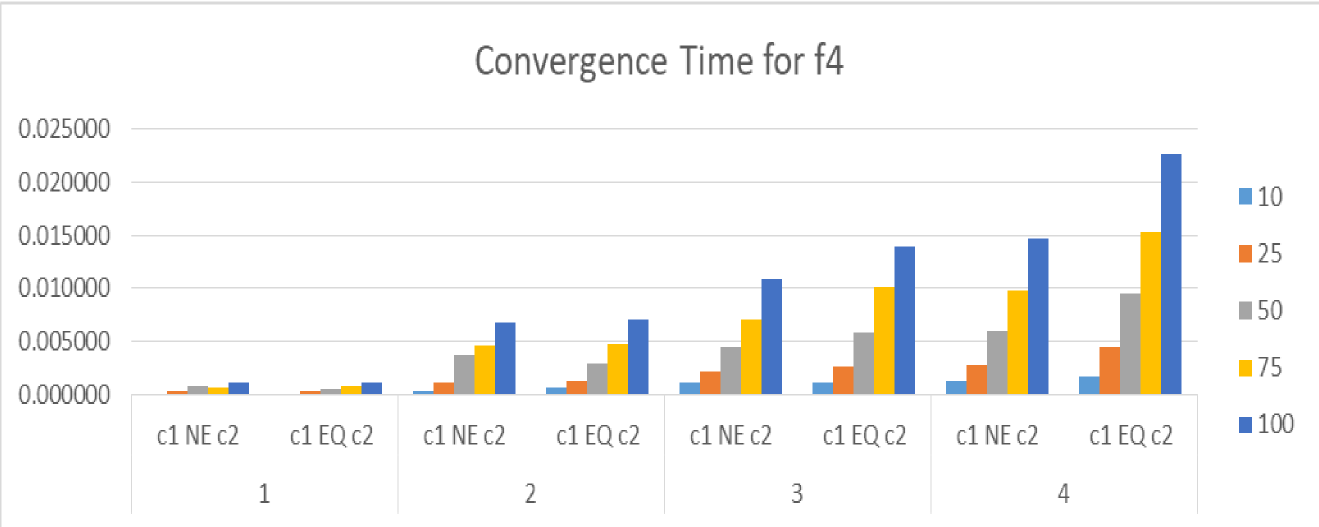

Fig. 9: Convergence of f4 (Dejong F4) benchmark function in seconds

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

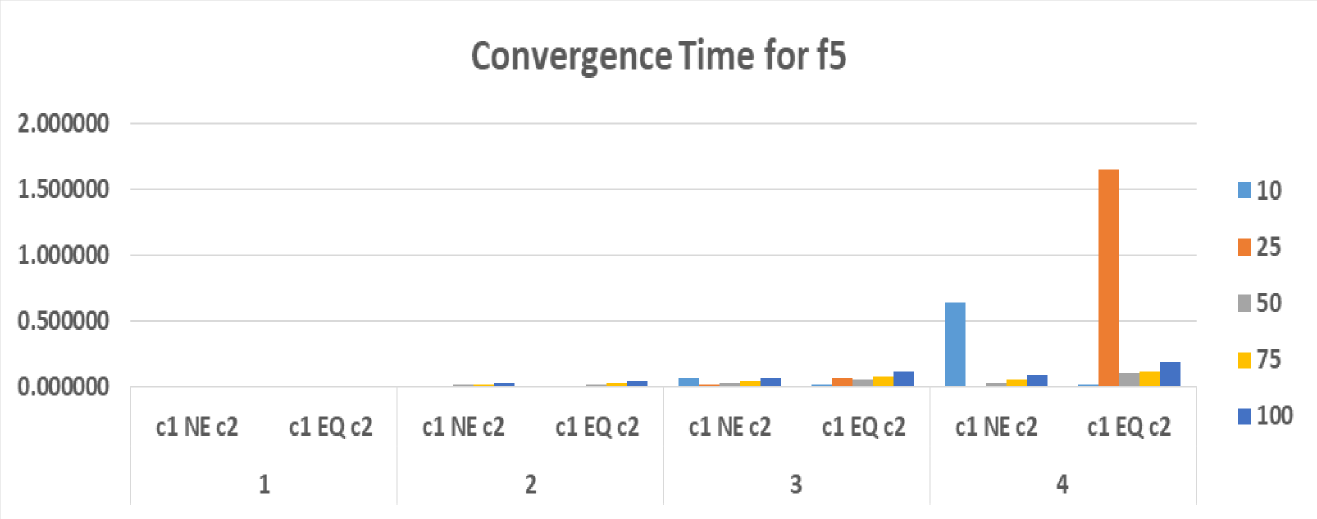

Fig. 10: Convergence of f5 (Alpine) benchmark function in seconds

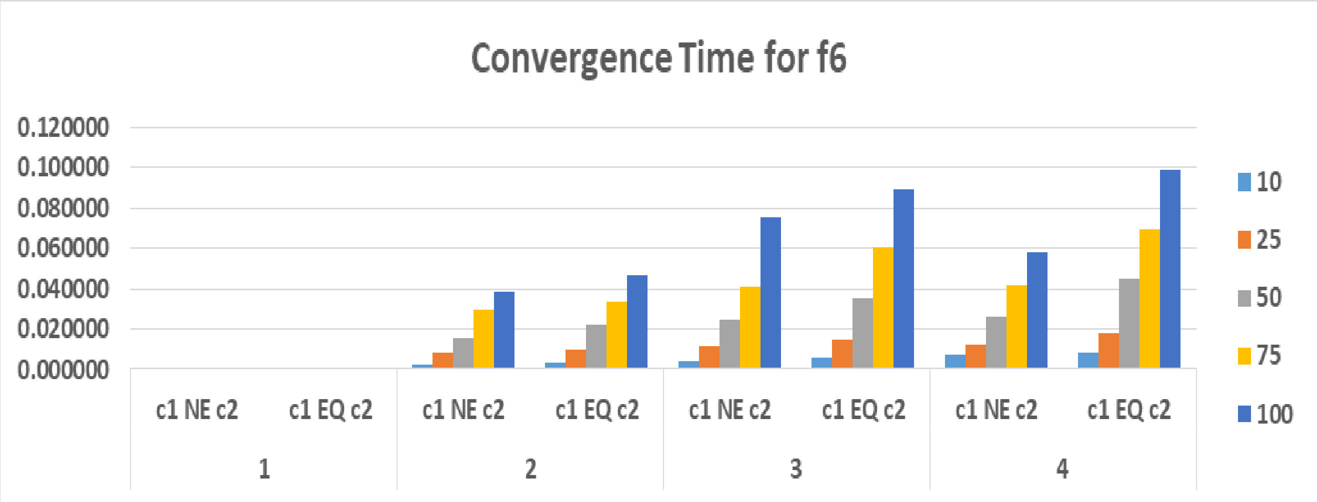

Fig. 11: Convergence of f6 benchmark function in seconds

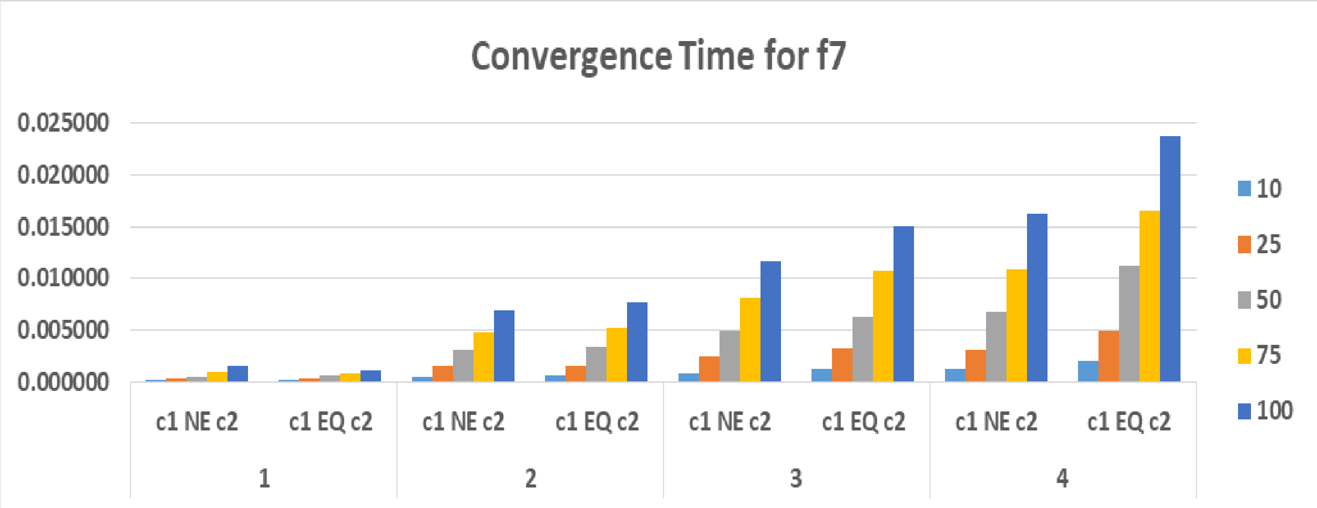

Fig. 12: Convergence of f7 (Quartic) benchmark function in seconds

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

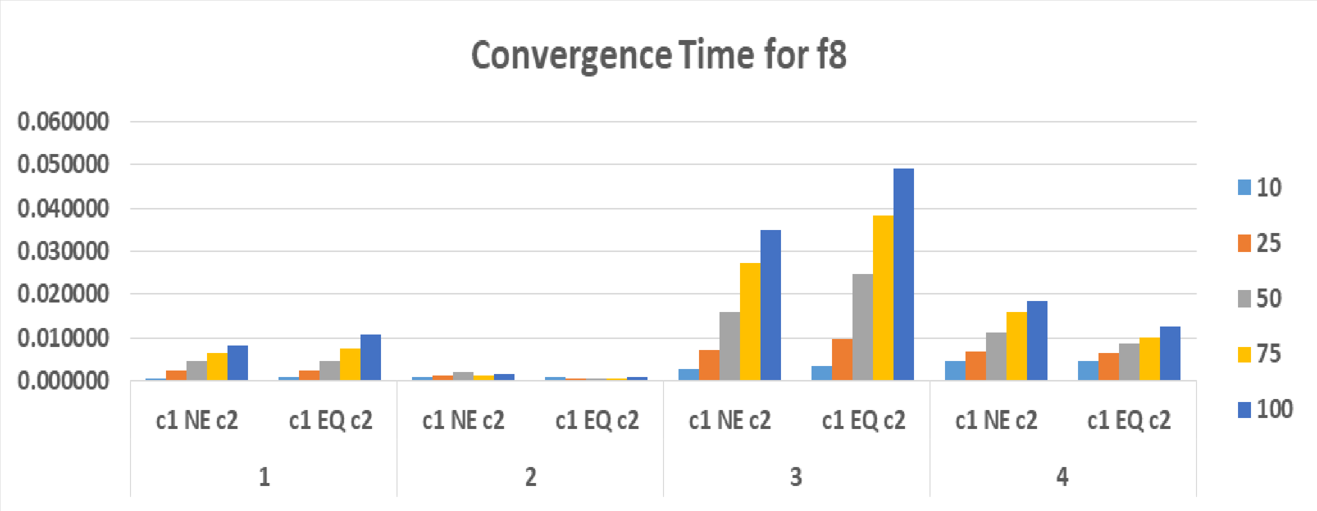

Fig. 13: Convergence of f8 (Schwefel 2.26) benchmark function in second

VI. CONCLUSION AND FUTURE WORK

The above study reveals that after BPSO, numerous researchers are working on improving the performance of BPSO with variant parameters like different initialization methods, Inertia weights, constriction factor, mutation operators, fuzzy logic, and parallelism in the computation of PSO. The results shows that variants are promising when compared with BPSO or their predecessors. The experimental study reveals that the performance of BPSO may be improved with further investigation on inertia weight and acceleration coefficients in various dimensions and various particle sizes.

REFERENCES

1. Craig W. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioural Model,” Computer Graphics, Vol.21, No.4, July, 1987, pp. 25-34.
2. James Kennedy and Russell Eberhart, “Particle Swarm Optimization,” In Proceedings of IEEE International Conference on Neural Networks, 1995, PP.1942-1948.
3. Riccardo Poli, “Analysis of the Publications on the Applications of Particle Swarm Optimisation,” Journal of Artificial Evolution and Applications, Vol.2008, January, 2018, 10 Pages.
4. Yudong Zhang, Shuihua Wang, and Genlin Ji, “A comprehensive survey on particle swarm optimization algorithm and its applications,” Mathematical Problems in Engineering, vol.2015, 2015, 38 pages.
5. Nguyen Quang Uy, Nguyen Xuan Hoai, Ri Mckay and Pham Minh Tuan, “Initialising PSO with randomised low-discrepancy sequences: the comparative results,” IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp.1985-1992.
6. H. Jabeen, z. Jalil and A.R Baig, “Opposition based initialization in Particle Swarm Optimization,” Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: late breaking papers, NY, USA, 2009, pp. 2047- 2052.
7. M. Pant, R. Thangaraj, C. Grosan, and A. Abraham, “Improved Particle Swarm Optimization with Low-Discrepancy Sequences,” in IEEE Cong. On Evolutionary Computing, Hong Kong, 2008, pp. 3011-3018.
8. M.G.H. Omran and Sal-Sharhan, “Using Opposition-based Learning to improve the performance of Particle Swarm Optimization,” IEEE Swarm Intelligence Symposium, 2008, PP 1-6.
9. Chang Zhang et al, “A Novel Swarm Model with Quasi-Oppositional Particle,” International Forum on Information Technology and Applications, 2009, pp 325-330.
10. Matteo Diez, Andrea Serani, Cecilia Leotardi, Emilio Fortunato Campana, Giovanni Fasano, and Riccardo Gusso, “Dense Orthogonal Initialization for Deterministic PSO: ORTHOinit+,” International Conference on Swarm Intelligence 2016, pp 322-330.
11. Moaath Shatnawi, Mohammad Faidzul Nasrudin, Shahnorbanun Sahran, “A new initialization technique in polar coordinates for Particle Swarm Optimization and Polar PSO,” International Journal on Advanced Science, Engineering and Information Technology, Vol. 7, No.1, pp 242-249.
12. K.E. Parsopoulos, M.N. Vrahatis, “Initializing the Particle Swarm Optimizer Using the Nonlinear Simplex Method,” Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, 2002, WSEAS Press: 216-221.
13. Shahzad F., Baig A.R., Masood S., Kamran M., Naveed N., “Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO),” Advances in Computational Intelligence. Advances in Intelligent and Soft Computing, vol 116, pp339-348.
14. M. Alhussein and S. I. Haider, “Improved Particle Swarm Optimization Based on Velocity Clamping and Particle Penalization,” 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), Kota Kinabalu, 2015, pp. 61-64.
15. J. Tang and X. Zhao, “An Enhanced Opposition-Based Particle Swarm Optimization,” WRI Global Congress on Intelligent Systems, Xiamen, 2009, pp. 149-153.
16. Silva, A., Neves, A., Costa, E., “Chasing the swarm: A predator prey approach to function optimization,” Proceedings of the MENDEL 8th International Conference on Soft Computing, Brno, Czech Republic (2002)
17. Xu Jun-jie and Xin Zhan-hong, “An extended particle swarm optimizer,” 19th IEEE International Parallel and Distributed Processing Symposium, Denver, CO, 2005, pp. 5.
18. Maurice Clerc and James Kennedy, “The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space,” IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, February 2002, pp 58-73.
19. S. Raha, T. Som and N. Chakraborty, “Constriction Factor Based Particle Swarm Optimization Applied to Reactive Power Dispatch in Transmission System,” Second International Conference on Sustainable Energy and Intelligent System (SEISCON 2011), July. 20-22, 2011, pp 335-339.
20. Lim, S., Montakhab, M. and Nouri, H., “A Constriction factor based particle swarm optimization for economic dispatch,” European Simulation and Modelling Conference (ESM2009), 2009, October 26-28th 2009, 8 pages.
21. Mauro Sebastián Innocente and Johann Sienz, “Particle Swarm Optimization with Inertia Weight and Constriction Factor,” International conference on swarm intelligence (ICSI 2011), 2011, pages 11.
22. Yuhui Shi and Russell Eberhart, “A Modified Particle Swarm Optimizer,” IEEE World Congress on Computational Intelligence, 2009, PP 66-69.
23. H-R LI and Y-L Gao., “Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation,” in Second International Conference on Information and Computing Science, Manchester , 2009, pp. 66-69.
24. Huang Chongpeng, Zhang Yuling, Jiang Dingguo and Xu Baoguo, “On Some Non-linear Decreasing Inertia Weight Strategies in Particle Swarm Optimization*,” Proceedings of the 26th Chinese Control Conference, Zhangjiajie, Hunan, China, 2007, pp. 570-753.
25. Y. Shi and R. C. Eberhart, “Fuzzy Adaptive particle Swarm Optimization,” Proceedings of the IEEE Congress on Evolutionary Computation, Seoul , South Korea, 2001, pp. 101-106.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

26. L. Zhang, H. Yu, and S. Hu, "A new approach to improve particle swarm optimization," Proceedings of the 2003 international conference on Genetic and evolutionary computation, 2003, pp. 134-139.
27. J Wei and Y.Wang, "A Dynamical Particle Swarm Algorithm with Dimension Mutation," IJCSNS International Journal of Computer Science and network Security, Vol. 6, pp 221-224.
28. M. Pant and T. Thangaraj, V.P. Singh, "Particle Swarm Optimization Using Gaussian Inertia Weight," International Conference on Computational Intelligence and Multimedia Applications, Sivakasi, Tamil Nadu , 2007, pp. 97-102.
29. X. Liu et al, "Particle Swarm Optimization with Dynamic Inertia Weight and Mutation," Third International Conference on Genetic and Evolutionary Computing, Guilin, 2009, pp. 620-623.
30. P. Yadmellat, S. M. A. Salehizadeh and M. B. Menhaj, "A New Fuzzy Inertia Weight Particle Swarm Optimization," International Conference on Computational Intelligence and Natural Computing, Wuhan, 2009, pp. 507-510.
31. J. Liao, Y. Liu, X. Zhu, T. Xu and J. Wang, "Niching Particle Swarm Optimization Algorithm for Service Composition," IEEE Global Telecommunications Conference - GLOBECOM2011, Kathmandu, 2011, pp. 1-6.
32. Hui Wang and Yong Liu, "A Hybrid Particle Swarm Algorithm with Cauchy Mutation," Proceedings of the 2007 IEEE Sarm Intelligence Symposium, 5 pages.
33. Hui Wang, Yong Liu, Sanyou Zeng, Hui Li and Change Li, "Opposition-based Particle Swarm Algorithm with Cauchy Mutation," IEEE Congress on Evolutionary Computation, Sept 2007, PP 4750-4756.
34. M. Pant, R. Thangaraj, and A. Abraham, "Particle Swarm Optimization Using Adaptive Mutation," 19th International Conference on Database and Expert Systems, 2008, pp. 519-523.
35. M. Pant, R. Thangaraj, V.P Singhand and A. Abraham, "Particle Swarm Optimization Using Sobol Mutation," First International Conference on Emerging Trends in Engineering and Technology, Nagpur, Maharashtra , 2008, pp. 367-372.
36. Xiaoling Wu, Xiaojuan Zhao , "Particle swarm optimization based on power mutation," ISECS International Colloquium on Computing, Communication, Control, and Management, Sanya, 2009, pp. 464 – 467.
37. M. Imran, H.Jabeen, M. ahmad, Q. ababs, w.Bangyal and Q. Ababs , "Opposition Based PSO and Mutation Operators (OPSO with Power Mutation)," 2nd International Conference on Education Technology and Computer, Shanghai, 2010,pp.V4-506 -508.
38. M. Imran, Z. Manzoor, S. Ali and Q. Ababs , "Modified Particle Swarm Optimization with Student T Mutation," International Conference on Computer Networks and Information Technology, Abbottabad, 2011, pp. 283 - 286.
39. L.Chen , "Particle Swarm Optimization with a Novel Mutation Operator," International Conference on Mechatronic Science, Electric Engineering and Computer, Jilin, 2011, pp. 970-973.
40. C. Liu, C. Ouyang, P. Zhu and W. Tang, "An Adaptive Fuzzy Weight PSO Algorithm," Fourth International Conference on Genetic and Evolutionary Computing, Shenzhen, 2010, pp. 8-10.
41. Bo Wang, GuoQiang Liang, ChanLin Wang and YunLong Dong, "A new kind of fuzzy particle swarm optimization FUZZY_PSO algorithm," 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, 2006, pp. 3 pp.-311.
42. H. Ji, J. Jie, J. Li and Y. Tan, "A Bi-swarm Particle Swarm Optimization with Cooperative Co-evolution," International Conference on Computational Aspects of Social Networks, Taiyuan, 2010, pp. 323-326.
43. G. Dashora and P. Awwal, "Adaptive particle swarm optimization employing fuzzy logic," International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016, pp. 1-4.
44. F. Olivas, F. Valdez and O. Castillo, "Particle swarm optimization with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions," World Congress on Nature and Biologically Inspired Computing, Fargo, ND, 2013, pp. 36-40.
45. S. Kumar and D. K. Chaturvedi, "Tuning of Particle Swarm Optimization Parameter Using Fuzzy Logic," International Conference on Communication Systems and Network Technologies, Katra, Jammu, 2011, pp. 174-179.
46. D. Yang, J. Chen and N. Matsumoto, "Particle Swarm Optimization with Adaptive Parameters," Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, 2007, pp. 616-621.
47. J. J. Liang, L. Guo, R. Liu and B. Y. Qu, "A self-adaptive dynamic particle swarm optimizer," IEEE Congress on Evolutionary Computation (CEC), Sendai, 2015, pp. 3206-3213.
48. Dazhi Wang, Dingwei Wang, Yang Yan and Hongfeng Wang, "An adaptive version of parallel MPSO with OpenMP for Uncapacitated Facility Location problem," Chinese Control and Decision Conference, Yantai, Shandong, 2008, pp. 2387-2391.
49. S. Baskar and P. N. Suganthan, "A novel concurrent particle swarm optimization," Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 2004, pp. 792-796 Vol.1.
50. Y. Fukuyama, "Parallel particle swarm optimization for reactive power and voltage control investigating dependability," 18th International Conference on Intelligent System Application to Power Systems (ISAP), Porto, 2015, pp. 1-6.
51. Y. D. Ping, Z. Kai, F. L. Bo and Z. Ming, "A Parallel Chaos Particle Swarm Optimization," International Conference on Environmental Science and Information Application Technology, Wuhan, 2009, pp. 645-648.
52. S. K. Singh and R. Kumar, "Scheduling in multiprocessor systems using parallel PSO," International Conference on Computing, Communication & Automation, Noida, 2015, pp. 175-180.
53. C. J. A. Bastos-Filho, D. F. Carvalho, E. M. N. Figueiredo and P. B. C. de Miranda, "Dynamic Clan Particle Swarm Optimization," Ninth International Conference on Intelligent Systems Design and Applications, Pisa, 2009, pp. 249-254.
54. A. McNabb, M. Gardner and K. Seppi, "An exploration of topologies and communication in large particle swarms," IEEE Congress on Evolutionary Computation, Trondheim, 2009, pp. 712-719.
55. Lin Lu, Qi Luo, Jun-yong Liu and Chuan Long, "An improved particle swarm optimization algorithm," IEEE International Conference on Granular Computing, Hangzhou, 2008, pp. 486-490.
56. S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer, The 2003 Congress on Evolutionary Computation," 2003. CEC '03, Canberra, ACT, Australia, 2003, pp. 770-776 Vol.2.
57. J. Ren and S. Yang, "An Improved PSO-BP Network Model," Third International Symposium on Information Science and Engineering, Shanghai, 2010, pp. 426-429.
58. M. Alam, S. Chatterjee and H. Banka, "A novel parallel search technique for optimization," 3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, 2016, pp. 259-263.
59. D. Tian and N. Li, "Fuzzy Particle Swarm Optimization Algorithm," International Joint Conference on Artificial Intelligence, Hainan Island, 2009, pp. 263-267.
60. J. C. Vazquez and F. Valdez, "Fuzzy logic for dynamic adaptation in PSO with multiple topologies," Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Edmonton, AB, 2013, pp. 1197-1202.
61. Momin Jamil and Xin-She Yang, "A literature survey of benchmark functions for global optimization problems," Int. Journal of Mathematical Modelling and Numerical Optimisation, Vol. 4, No. 2, pp. 150–194 (2013).

AUTHORS PROFILE

YVR Naga Pawan, pursing Ph.D. in the Department of Computer Science and Engineering in Koneru Lakshmaiah Education Foundation, Vijayawada, A.P., INDIA. The areas of interests are machine learning, Databases, Network Security.

Kolla Bhanu Prakash, is Ph.D. from Satyabhama University and working as Professor in the Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vijayawada, A.P., INDIA. The areas of interests are Machine Learning, Knowledge Engineering.