THE BUTTERFLY-PARTICLE SWARM OPTIMIZATION (BUTTERFLY-PSO/BF-PSO) TECHNIQUE AND ITS VARIABLES

[1]Aashish Kumar Bohre, [2]Dr. Ganga Agnihotri and [3]Dr. Manisha Dubey

[1,2,3]Deptt. of Electrical Engg, MANIT, Bhopal, MP, INDIA

ABSTRACT

The new presented Butterfly-PSO technique (or BF-PSO) is basically originated by Particle Swarm Optimization (PSO). The Butterfly-PSO technique (BF-PSO) appears as a new growing star among all optimization techniques. The proposed 'Butterfly- Particle Swarm Optimization (Butterfly or BF-PSO)' is inspired by butterfly natural intelligence, character, behavior, intelligent network and intelligent communication during the nectar search process. The BF-PSO introduces new parameters such as sensitivity of butterfly (s), probability of food (nectar) (p), the degree of the node and the time varying probability coefficient (α). These parameters improve the searching ability, excellent convergence and the overall performance of the Butterfly-PSO effectivly. The BF-PSO optimizations results have been presented for various functions with the multi-dimension problems.

KEYWORDS

Optimization techniques, Particle swarm optimization (PSO), Butterfly-PSO (BF-PSO), Butterfly communication network, Sensitivity, Probability of nectar.

1. INTRODUCTION

The Artificial intelligence (AI) based optimization techniques employs a variety of biological phenomena, which helps for implementation and computation. The artificial intelligence technique uses many fundamentals associated with the life. Within the last few years artificial intelligence approach has deployed various efficient methodologies to solve complexity in the problems. These techniques investigate the best solution in the certain search space by utilizing best experiences and knowledge. The initial artificial systems utilize analogy with the Darwin's theory and it's the basic principle "survival of the fittest" [1].

The implementation of artificial techniques doesn't follow whole natural system; but it's explores and identifies the ideas, also implement and model. The different insects have capability to perform variety of typical jobs and process. The frequently organized example in insects is the collection of food and process for it. The swarms of various insect such as birds, bees, butterflies, ants are the best examples of this interesting behavior in nature [3]. The important aspect in insect swarm is the interactions between each other particles, effects of inter-particles and search capability analysis. The collective intelligence behavior in the swarm intelligent systems is important phenomenon, by which the agents interacts with each other in the search environment. The learning ability and particle intelligence plays an important role in food searching process for

DOI : 10.14810/ijscmc.2015.4302

different type of insects. The demonstrations for the essential flower foraging in taxa such as butterfly, bees, parasitoid wasps, butterflies and moth [3,5,6]. The Butterflies showed a significantly higher learning rate, capability and intelligence than other insects [6-7].

The basic concept of genetic based evolutionary search algorithm known as Genetic Algorithm (GA); which is motivated by the natural adaptations of the biological species and follows the concept of Darwinian's theory presented by Holland [1]. The applications of Genetic Algorithms (GAs), Particle swarm optimization (PSO), and artificial bee colony (ABC) algorithm for complex functions has proposed by Haupt et al. [2]
.
The study of Swarm Intelligence and comprehensive models of the behaviors and intelligence of the social insects as well as the way to utilize this kind of models within the particular pattern or design associated with complicated system has given by Bonabeau et al. [3]. The nonlinear functions optimized by using particle swarm optimization methodology, whose concept was given by Kennedy and Eberhart [4, 5] and the different benchmark functions have analyzed considering with several paradigms.

A.K. Bohre et al. [6-7] have given the concept of Butterfly-particle swarm optimization (Butterfly-PSO or BF-PSO) technique based on the characteristic behavior intelligence and the butterfly swarm search process for food hence attraction towards food (or nectar) source. They have included several modern parameters such as sensitivity, probability etc. The motivation towards the butterfly based swarm optimization is searching of food processing, intelligence and behavior. The searching process of butterflies basically concentrate on the food source that is nectar sources. The butterflies have the natural sensitivity to sense the nectar probability. Butterfly develops an interactive intelligent system with high communication to find the optimal solutions.

Babayigit et al. [8] tested the modified artificial bee colony (ABC) algorithm for the numerical optimization problems, different probability function and the new searching mechanism were presented to improve the performance of it. Demongeot et al. [9] studied the impact of fixed boundary conditions including with the attraction towards flower plants in real biological networks and also giving an example of boundary influence in the genetic regulatory network of the flower's morphogenesis of the plant Arabidopsis thaliana. Leung et al. [10] considered the quantum communication between specific sender-receiver pairs (two-way) utilizing with the butterfly network structure. Mercader et al. [11] have analyzed the antennal sensitivity between insect and plants. This includes a strong specialization in the olfactory system for the butterfly species. Lucas et al. [12] investigated the broad functional diversity with the ability to detect sounds across a wide range of frequencies and intensities of insects and also the remarkable variation in structure is associated with function that provides a selective advantage, particularly in predator detection. Cunningham et al. [13] provided the information about the plant-insect (or flower-butterfly) communication and the learning behavior for foraging with deciding the preference base on odour as well as floral volatility. M. Erik et al. [14] have compared test results between PSO (Particle Swarm optimization) and MOL (Many Optimizing Liaisons) including with the several benchmark functions problems and the numerical results of optimizations. J.C. Bansal et al. [15] has given the concept for variations in Inertia Weight based on that the PSO can modify in many ways with the different functions.

The proposed work projects a novel optimization technique and its variables based on the Butterfly-particle swarm optimization for the several benchmark functions and considering the complex problems for many dimensions. By considering the new parameters; overall performance, superiority, searching ability and the convergence of optimization technique have enhanced than all available techniques.

2.THE BUTTERFLY ARTIFICIAL INTELLIGENT NETWORK

The butterfly is an advance insect having a complete life cycle with four different stages. These stages are eggs, caterpillar, chrysalis, adult butterfly [16]. The adult butterflies have colorful wings, maturity to sense, long branches, compound eyes and long antenna. The adult butterfly food search process and behaviors are more realistic for finding the nectar sources or food plants (flowers). In the food search process the butterflies find nectar sources or food plants (flower nodes). The butterflies have different sensors such antenna, eyes, etc., to find out the food plant. The butterfly can communicate or exchange information between them and neighbors also by various ways, such as dancing, color, chemicals, sound, and physical actions, etc. By this butterfly shows the collective intelligence behavior on the butterfly network [6].

The communication network graph of butterfly has originally taken from the graph of FFT networks, which can execute the fast Fourier transform (FFT) very effective way. The butterfly network graph forms by implementing the wings structure in to the graph theory. The interconnected network between two or more butterflies is represented by the cascaded connection of wings structure. The butterfly intelligence network (based on wing structure) is used to represent the linear network also. This butterfly network having different nodes A, B, C and D as shown in figure-1. These vertex or node can be assumed as communication nodes of butterfly network, which is used to exchange or transmit information between butterflies by different means [6,7].

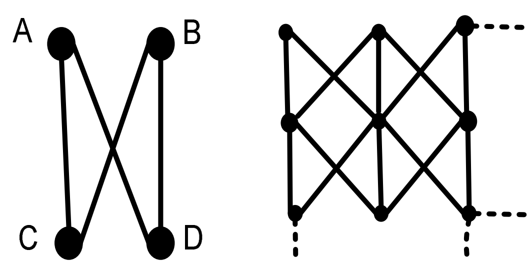


Figure-1: The Butterfly Communication Network Structure [6]

The artificial butterfly intelligence network formed in the sensitive and probabilistic region minimum sensitivity of butterfly meets with minimum probability of nectar. The nodes in sensitive and probabilistic region are said to be active state and out of this region butterfly is not able to detect the probability of nectar so its called in the inactive state. The active and inactive regions are separated by the boundary which is formed when the minimum sensitivity of butterfly meets to the minimum probability of nectar. Let us assume that a network having N nodes i.e. sources of nectar. The nodes are selected on the basis of its degree. If the node doesn’t have nectar then it cannot be sense by butterfly hence the degree of node will be zero and if the node (nectar source or flower) having the nectar then based on the sensing ability of butterfly connectivity of node will increase hence the degree increase (higher degree). The optimal solution in the sensitive and probabilistic region will depend on the degree of nodes. The node (flower location or position) having more amounts of nectar in the butterfly network, will attract more and more butterflies by their natural sensitivity, communication and natural intelligence. And hence; it will increase the degree of the node (flower) because that node has higher probability of nectar [6-7]. The butterflies exchange the information by different way of communications by utilizing natural intelligences such as dancing, sensing nectar, colors, insects, chemicals, sounds and many physical actions [9, 10, 11, 12, 13]; hence better communication network is developed by butterfly as compared to insects. The butterflies form the "collective intelligence" by following the intelligence communication network and the different types of communicated information's spread between the individual butterflies and neighbors. In process for calculating the optimal solution, the degree of node in every flight of butterfly is assumed as approximately equal to 1 because assuming maximum connectivity in each flight. The states of node configuration of the butterfly network for a particular instant in general are given by equation (1) as:

$$\boldsymbol{s_k}\ and\ \boldsymbol{p_k} = \begin{cases} = 0; & inactive \\ > 0; & active \end{cases} \tag{1}$$

3.THE BUTTERFLY SWARM SEARCH PROCESS FOR OPTIMAL SOLUTION

The butterfly swarm based search process investigates the optimal location depending upon the sensitivity of flower and probability of nectar. The information about the optimal location solution communicates directly or indirectly in all butterflies by different ways of communication intelligence (such as dancing, colors, chemicals, sounds and physical actions). The intention of butterfly selects the flower having appropriate nectar probability with its sufficient sensitivity in random search for the optimal solution and before terminating the respective flight (iteration) it intended process towards next optimal solution. This search process will continue till the convergence or termination criteria is achieved. During this search the butterflies are able to choose several local best or *lbest* positions, sensitivity and probability they move towards global best or *gbest* position (or location). The butterfly decision making process in the search region depends on the butterfly interaction with the substantially different sensory system [6, 7, 11, 12, 16]. The termination criteria will be the convergence and maximum number of flight (iteration) along with improvement in respective fitness. The butterfly-PSO according to the intelligence characteristics, behaviors and due to antennal on their mouth by which flowers attract butterfly more compare to others [6, 11, 12]. There is very good relation between the butterfly and the surrounding atmosphere to provide optimal solution [7, 12, 16]. In the search process butterfly moves towards the new nectar source, just after taking the food (or nectar).

The figure-2 gives the representation of the butterfly-PSO search process for *gbest* solution considering with the flower node from N_1 to N_6 and shows the movement of butterfly from i^{th} location to $i+1^{th}$ location during the search process of BF-PSO. The flying decision for each next flight in the search process is made based on the butterfly particle swarm's previous history (best experience of self and neighbors). Based on the history of the best experience of self and neighborhood of swarms the velocities and positions are modified. The sensitivity and probability concepts follow the survival of the fittest principle like other evolutionary techniques. The local best (*lbest)* position of a butterfly which has the more sensitivity of butterfly and most probability of nector will become global best (*gbest*) solution.

4. THE BUTTERFLY-PSO (BF-PSO) TECHNIQUE IMPLEMENTATION

The PSO developed by Kennedy and Eberhart (1995) is a well-known optimization algorithm [5]. Throughout the PSO, every particle within the search space follows a particular velocity and inertia with the related generations and updates its positions. Let us assume total N positions (or populations) vector x (i=1, 2, 3….N) and respective velocities vector v. Consider v_k and x_k are the velocity and population respectively for k^{th} iteration and also v_{k-1} and x_{k-1} velocity and population respectively for previous iteration. Hence based on basic-PSO the equations (equations 2 and 3) for velocity and position (or population) for kth iteration can be given as [4, 5, 15]:

$$v_k = w_k * v_{k-1} + c_1 r_1 (x_{pbest,k-1} - x_{k-1}) + c_2 r_2 (x_{gbest,k-1} - x_{k-1}) \quad (2)$$

$$x_k = x_{k-1} + v_k \quad (3)$$

Where, c_1 & c_2 are acceleration coefficients and r_1& $r_2$are random variable (0 to 1).

The butterfly leaning based particle swarm optimization algorithm has developed to find the optimal solutions including the random parameters, acceleration coefficients, probability, sensitivity, lbest and gbest for the fast convergence and more accurate solutions than other methodology. In the Butterfly-PSO, lbest solutions are selected by the individual's best solution. After that the gbest solution identified based on the respective fitness. The locations (position) of the nectar (food) source represent the probable optimal solution for the problem and the amount of nectar (food) represents the corresponding fitness. The detail implementation of the Butterfly-PSO (BF-PSO) technique is given below and the search process for Butterfly–Particle Swarm Optimization (BF-PSO) representation is given in figure-2. The general ranges of the sensitivity and probability are considering from 0.0 to 1.0. The velocity limits can be set based on the limits of the problem variables.

Hence the function of sensitivity and probability as a function of iterations can be given as:

$$s_k = exp\text{-}(ITER_{max} - ITER_k)/ ITER_{max} \quad (4)$$

Where, $ITER_{max}$ = maximum number of iterations, and $ITER_k$ = k^{th} iteration count.

$$p_k = FIT_{gbest,k} / \sum(FIT_{lbest,k}) \quad (5)$$

Where $FIT_{lbest,k}$ =Fitness of local best solutions with k^{th} iteration, $FIT_{gbest,k}$ = Fitness of global best solutions with k^{th} iteration.

The values of the acceleration coefficients c_1 and c_2 are equals to 2 for BF-PSO and as well as for the conventional or standard- PSO. The inertia weight range for both BF-PSO is 0 to 1 and which can be given as:

$$W_k = (ITER_{max} - ITER_k)/ ITER_{max} \quad (6)$$

The basic concept of velocity-displacement-time is:

$$Velocity(\mathrm{v}) = Displacement(\mathrm{x}) / Time(\mathrm{t}) \quad (7)$$

By taking the $k_1=1/Time(t)$; that is assume that the random time constant (k_1) for any instantaneous speed and distance.

$$Velocity(\mathrm{v}) = Displacement(\mathrm{x}) \times k_1 \quad (8)$$

Also the velocity is directly proportional to the inertia and hence different velocity values have different inertia. The thumb rule for summation is that, the summation can possible for similar or equivalent similar quantities. So, the general equation for updating the velocity is given as:

$$Velocity' = Displacement_1 \times k_1^{'} + Displacement_2 \times k_1^{''} + Inertia \times Velocity \times k_1^{'''} \quad (9)$$

Similarly following the thumb rule the location displacement can updated follows, if $k_2' = Time(t)$ is random instant than,

$$Displacement' = Displacement + Velocity \times k_{2}^{'} \quad (10)$$

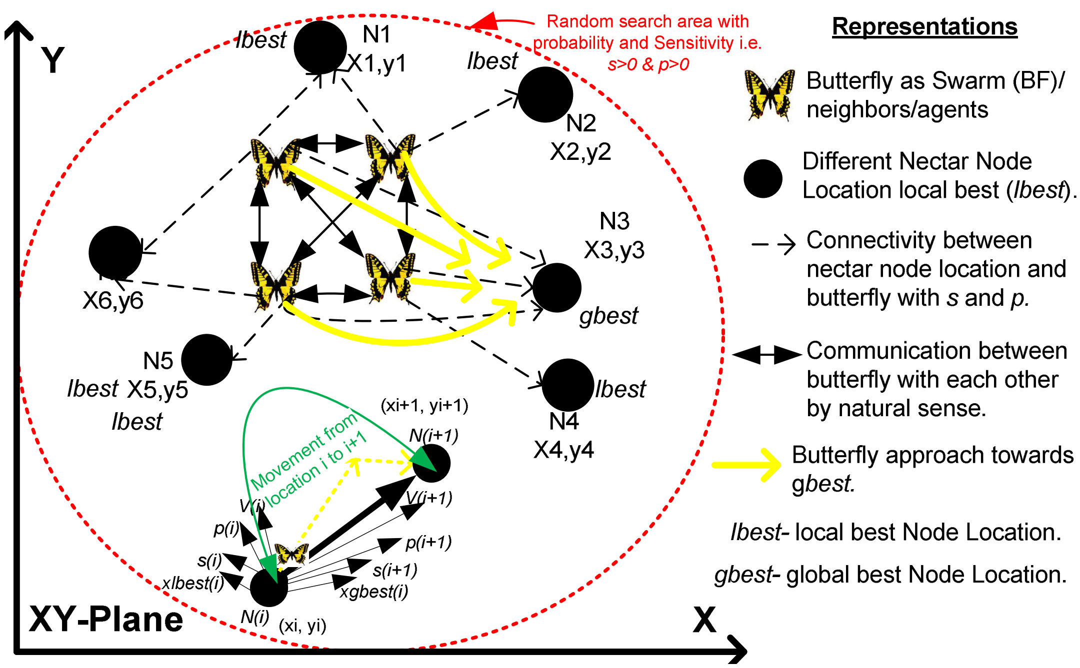


Figure-2: The search process for Butterfly–Particle Swarm Optimization (BF-PSO) representation

Now by applying the Butterfly- Particle Swarm Optimization (Butterfly or BF-PSO) in above concept, the butterfly velocity and location (or position) depends on the sensitivity of butterfly and probability of nectar amount in the search. Hence the equations for updating the velocity and location (or position) are the function of the sensitivity and probability as given in equations 11 and 13. The figure-3 gives the flow chart for Butterfly-Particle Swarm Optimization.

Now the final equations for the Butterfly-PSO (BF-PSO) can be given in equations 12 and 13, with all factors as:

$$v'_k = k_1^{'''}.w_k.v_{k-1} + s_k(1-p_k)c_1r_1(x_{lbest,k-1} - x_{k-1}) + p_{kg}c_2r_2(x_{gbest,k-1} - x_{k-1}) \qquad (11)$$

Let us assuming $k_1^{'''} = 1$, then

$$v'_k = w_k.v_{k-1} + s_k(1-p_k)c_1r_1(x_{lbest,k-1} - x_{k-1}) + p_kc_2r_2(x_{gbest,k-1} - x_{k-1}) \qquad (12)$$

And,

$$x_k = x_{k-1} + \alpha_k.v'_k \qquad (13)$$

Where, p_{kg} is the probability of global best (generally assume p_{kg}=1 for global solution), , p_k is the currnt probality (as given in equation no. (5)) and α_i is time varying probability coefficient, α_k=*rand**p_k, *rand*-is the random number [0, 1].

5.THE VARIABLES OF BUTTERFLY-PSO (BF-PSO)

The Butterfly or BF-PSO updates the velocity and location according to equations 11 and 13. These equations included the many variables such as:

- Sensitivity (s).
- Probability (p).
- The time varying probability coefficient (α).
- The acceleration coefficients (c1 and c2).
- The inertia weight (w).
- The variable $k_1^{'''}$ (in this work assumed equals to 1).
- The degree of node (assumed equals to 1).

The sensitivity (s), probability (p), time varying probability coefficient(α), acceleration coefficients (c1 and c2), variable $k_1^{'''}$ and the inertia weight (w) are not the fix function but these are variables, which can be given by the some appropriate mathematical function expressions to modify the methodology and results. The functions for these variables can be given in efficient manner by different exponential functions, logarithmic functions, combinational or some combined functions, simple iteration based functions, iteration based exponential functions, iteration based logarithmic functions and combinations of given functions etc. Similarly these variables can be adopted in many ways by researchers to improve results and modify the methodology.

6.THE RESULTS ANALYSIS

The BF-PSO is applied on the several well known standard benchmark functions for the validation of results. The results of proposed methodology are determined with MATLAB (2009a) and for minimization of all the objective functions with the system configurations windows-8.1, AMD-E1-1500APU, 1.48 GHz, 2.0 GB RAM. The parameters value for the BF-PSO optimization techniques are given in table-1. The benchmark functions and the ranges for these are given in table-2. The results with minimum value, mean value and standard deviation value for the benchmark functions are given in table-3. The global best variables (positions or dimensions) values for respective minimum value of fitness value are given in table-4 and the MATLAB coding for BF-PSO technique is given in appendix-I.

The result in table-3 gives the values of minimum value, mean value and standard deviation value of fitness for all the benchmark functions. The table result includes the best results for minimum fitness, mean fitness and standard deviation in 100 trials. The iterative convergence analysis of minimum fitness for Sphere, Ackley, Rastrigin, Rosenbrock, Griewank and Schwefel-12 functions have given in figure-4. The obtained values of global best parameters with respective minimum fitness are given in table-4. The BF-PSO result analysis for Sphere function the values of minimum fitness, mean fitness and standard deviation for 30 dimensions are 0, 47.9053 and 1.5155e+003 respectively, similarly the comparative results for Ackley function are 8.8818e-016, 0.0238 and 0.6431, Rastrigin function are 0 0.3924 and 11.7406 , Rosenbrock function are 28.9288 and 0.9316, Griewank function are 0, 0.4115 and 12.9878, and Schwefel-12 function are 0, 45.5966 and 1.4424e+003. Also the analysis of other dimensions (i.e. D = 2, 20) can be done for different benchmark function. The BF-PSO obtains the best values results within the minimum number of iteration, so it shows the higher convergence rate, increased diversity and better accuracy in the BF-PSO techniques for the benchmark functions, which are given in figure 4.

Table-1: The parameters value for BF-PSO optimization technique

Number of trials	100
Maximum no. of iteration	1000
The butterfly swarm size	20
c_1	2
c_2	2
Inertia weight (w)	0.0 → 1.0
Probability factor (p)	0.0 → 1.0
Sensitivity factor (s)	0.0 → 1.0

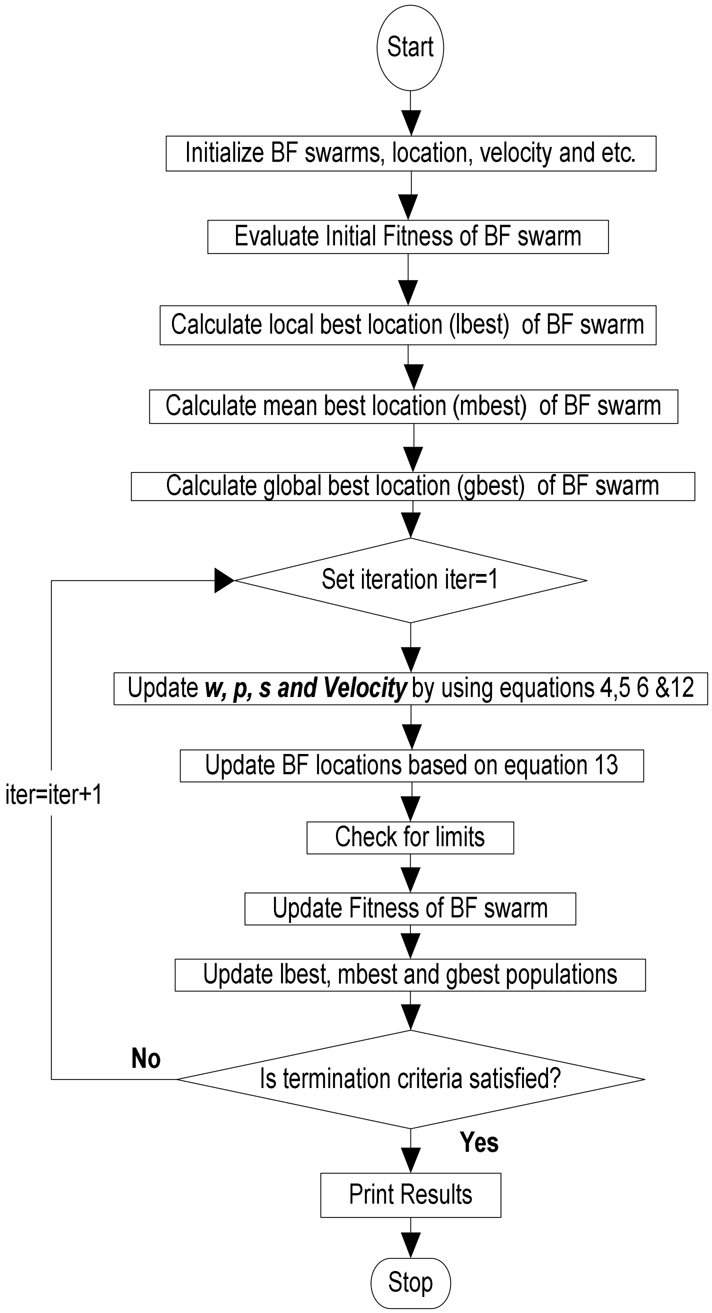
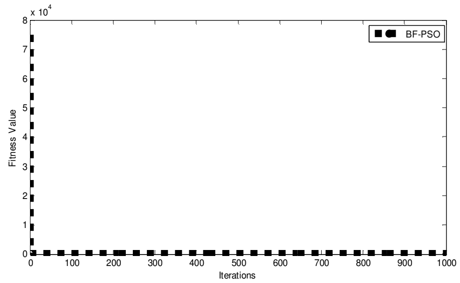
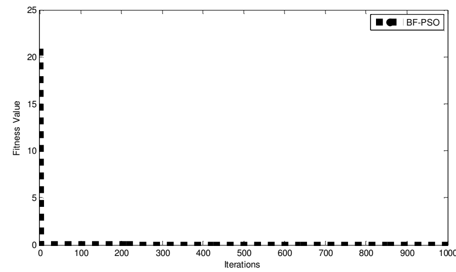


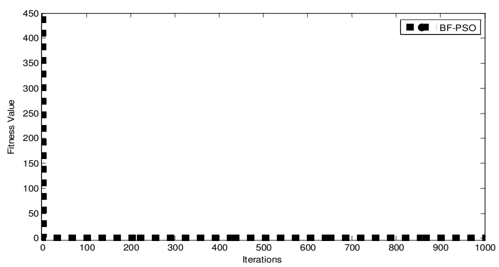
Figure-3: The Flow chart for Butterfly-Particle Swarm Optimization (BF-PSO) algorithm



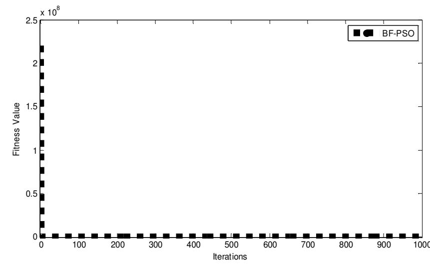
(a) Sphere function



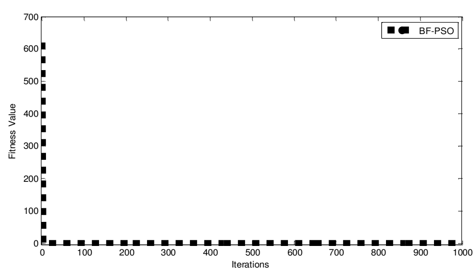
(b) Ackely function



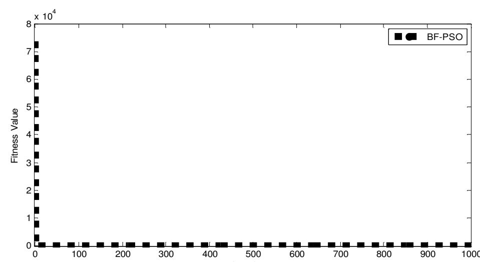
(c) Rastrigin function



(d) Rosenbrock function



(e) Griewank function



(f) Schwefel 12 function

Figure-4: The BF-PSO results for minimization of benchmark functions (a,b,c,d,e,f) with 30 dimensions.

7.CONCLUSIONS

The BF-PSO technique has given the better results in all respects as discussed above. The graphic and numeric result indicates that the Butterfly-PSO (or BF-PSO) is a new growing and shining star in the fields of optimization techniques. The main conclusions related to Butterfly-PSO (or BF-PSO) can be drawn as:

- Improves the searching ability.
- Excellent convergence.
- Higher accuracy.
- It ignores the problem of premature convergence.
- It reduces time taken for final value conversion.
- It takes less iteration for final value conversion.

REFERENCES

[1] J. H. Holland, *"Adaptation in Natural and Artificial Systems"* Ann Arbor: The University of Michigan Press, 1975.

[2] R. L. Haupt and S. E. Haupt, *"Practical Genetic Algorithms"* second edition, 2004, published by John Wiley & sons, inc., hoboken, new jersey, published simultaneously in Canada, 2004.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, *"Swarm intelligence: From natural to artificial systems"* Oxford University Press, 1999.

[4] J. Kennedy, and R.C. Eberhart, *"Particle Swarm Optimization"* IEEE International Conference on Neural Networks Proceedings, pp. 1942-1948, vol. 4, Nov. - Dec. 1995. doi: 10.1109/ICNN.1995.488968.

[5] R.C. Eberhart, and J. Kennedy, *"A new optimizer using particle swarm theory"* Proceedings Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, pp.39-43, 4-6, Oct, 1995. doi: 10.1109/MHS.1995.494215.

[6] A.K. Bohre, G. Agnihotri, and M. Dubey, *"Hybrid butterfly based particle swarm optimization for optimization problems"* First International Conference on Networks & Soft Computing (ICNSC), 2014, pp.172-177, 19-20 Aug. 2014. doi: 10.1109/CNSC.2014.6906650.

[7] A.K. Bohre, G. Agnihotri, M. Dubey, and J. S. Bhadoriya *"A novel method to find optimal solution based on modified butterfly particle swarm optimization"* International Journal of Soft Computing, Mathematics and Control (Wireilla-IJSCMC), pp.1-14, vol. 3, no. 4, Nov. 2014. doi: 10.14810/ijscmc.2014.3401.

[8] B.K. Babayigit, R. Ozdemir , *"A modified artificial bee colony algorithm for numerical function optimization"* IEEE Symposium on Computers and Communications (ISCC), pp. 000245 – 000249, IEEE , 2012.

[9] J. Demongeot, M. Morvan and S. Sene, *"Impact of Fixed Boundary Conditions on the Basins of Attraction in the Flower's Morphogenesis of Arabidopsis Thaliana"* IEEE, 22nd International Conference on Advanced Information Networking and Applications, Workshops, WAINA, 2008, pp. 782-789.

[10] D. Leung, J. Oppenheim, and A. Winter *"Quantum Network Communication-The Butterfly and Beyond"* IEEE Transactions on Information Theory, vol. 56, no. 7, pp. 3478-3490, July 2010.

[11] R. J. Mercader, L. L. Stelinski and J. M. Scriber, *"Differential antennal sensitivities of the generalist butterflies papilio glaucus and p. canadensis to host plant and non-host plant extracts"* Journal of the Lepidopterists' Society, 62(2), pp. 84-88, 2008.

[12] K. M. Lucas, J. F. C. Windmill, D. Robert and J. E. Yack, *"Auditory mechanics and sensitivity in the tropical butterfly Morpho peleides (Papilionoidea, Nymphalidae)"* The J. Experimental Biology, 212, pp. 3533-3541, July 2009. doi:10.1242/jeb.032425

[13] J. P. Cunningham, C. J. Moore, Myron P. Zalucki and Stuart A. West *"Learning, odour preference and flower foraging in moths"* The J. Experimental Biology, 207, pp. 87-94, September 2003. doi: 10.1242/jeb. 00733

[14] M. Erik, H. Pedersen, H. Laboratories, *"Good Parameters for Particle Swarm Optimization"* Technical Report, no. HL1001, pp. 1-12, 2010. http://www.cof.orst.edu/cof/teach/fe640/Class_Materials/Particle%20Swarm/ PSO%20parameters.pdf

[15] J.C. Bansal; P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, A. Abraham, *"Inertia Weight strategies in Particle Swarm Optimization"* Third World Congress on Nature and Biologically Inspired Computing (NBIC), 2011, pp.633,640, 19-21 Oct. 2011. doi: 10.1109/NaBIC.2011.6089659

[16] http://www.ansp.org/explore/online-exhibits/butterflies/lifecycle/

AUTHORS

Aashish Kumar Bohre[1] received B.E. degree in Electrical and Electronics engineering from UIT- RGPV Bhopal, (2009), and M-Tech degree in Power System (2011) from MANIT, Bhopal. At the moment he is Ph.D. scholar at MANIT, Bhopal, India. His research interest includes Distribution system operation and planning, Distributed generation, Power System Optimization, Smart Grid Operation. Email: aashu371984@gmail.com, aashish_bohre@yahoo.co.in

Dr. Ganga Agnihotri[2] received BE degree in Electrical engineering from MACT, Bhopal (1972), the ME degree (1974) and PhD degree (1989) from University of Roorkee, India. Since 1976 she is with Maulana Azad College of Technology (currently MANIT), Bhopal in various positions. Currently she is professor. Her research interest includes Power System Analysis, Power System Optimization and Distribution Operation. Email: ganga1949@gmail.com

Dr. Manisha Dubey[3] was born in Jabalpur in India on 15th December 1968. She received her B.E (Electrical), M.Tech. (Power Systems) and Ph.D (Electrical Engg.) in 1990, 1997 and 2006 respectively. She is working as Professor at the Department of Electrical Engineering, Maulana Azad National Institute of Technology, Bhopal, India. Her research interests include power systems, Genetic Algorithms, Fuzzy Logic systems and application of Soft Computing Techniques in power system dynamics and control. Email: manishadubey6@gmail.com

Table-2: The benchmark function and ranges

Name	Functions	Range
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100, 100]
Ackley	$f_2(x) = -20\exp(-0.2 * \sqrt[2]{(1/n\sum_{i=1}^{n} x_i^2)}) - \exp(1/n\sum_{i=1}^{n}\cos(2\pi x_i)) + \exp(1) + 20$	[-32, 32]
Rastrigin	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10n)$	[-5.12, 5.12]
Rosenbrock	$f_4(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-30, 30]
Griewank	$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]
Schwefel 12	$f_6(x) = \sum_{i=1}^{n}\left(\sum_{i=1}^{n} x_i^2\right)^2$	[-100, 100]

Table 3: The comparative results for minimum fitness value of benchmark functions (1000 iteration and 100 trials)

Method → Functions ↓	D	BF-PSO (Fitness)		
		Minimum	Mean	Standard deviation
Sphere	2	0	0.0059	0.1864
	20	0	25.1406	795.3682
	30	0	47.9053	1.5155e+003
Ackley	2	8.8818e-016	0.0028	0.0859
	20	8.8818e-016	0.0228	0.5643
	30	8.8818e-016	0.0238	0.6431
Rastrigin	2	0	6.9082e-004	0.0218
	20	0	0.2407	7.4472
	30	0	0.3924	11.7406
Rosenbrock	2	1.9919e-029	0.1469	0.0387
	20	0.0111	1.9203e+005	2.7555e+006
	30	13.9382	4.5487e+005	6.4778e+006
Griewank	2	0	7.6443e-004	0.0242
	20	0	0.2465	7.7625
	30	0	0.4115	12.9878
Schwefel 12	2	0	0.0044	0.1394
	20	0	23.7908	752.6625
	30	0	45.5966	1.4424e+003

Table 4: The comparative results for global best positions with different dimensions of benchmark functions (1000 iteration and 100 trials)

Dim (n)	BF-PSO (gbest position)					
	Sphere	Ackley	Rastrigin	Rosenbrock	Griewank	Schwefel 12
2	0	0.2295e-015	0	1.0000	0	0
	0	-0.1617e-015	0	1.0000	0	0
20	-0.1046e-161	-0.1174e-015	0	0.9998	0	0.0149e-162
	-0.0013e-161	-0.0063e-015	0	0.9997	0	0.4097e-162
	-0.1065e-161	0.1353e-015	0	0.9995	0	0.3739e-162
	-0.0281e-161	0.1127e-015	0	0.9995	0	0.0029e-162
	0.0678e-161	0.0759e-015	0	0.9995	0	0.6909e-162
	0.0555e-161	0.0843e-015	0	0.9992	0	0.0141e-162
	0.0171e-161	-0.0787e-015	0	0.9995	0	-0.1157e-162
	-0.0103e-161	-0.1074e-015	0	0.9995	0	0.3685e-162
	0.0187e-161	0.0157e-015	0	0.9996	0	-0.0124e-162
	0.0874e-161	-0.0639e-015	0	0.9994	0	0.7854e-162
	-0.0121e-161	-0.0196e-015	0	0.9994	0	0.1670e-162
	-0.0129e-161	-0.0058e-015	0	0.9991	0	-0.4411e-162
	-0.0278e-161	-0.1263e-015	0	0.9984	0	-0.7398e-162
	0.0079e-161	0.0124e-015	0	0.9968	0	-0.4150e-162
	-0.0994e-161	-0.0627e-015	0	0.9939	0	-0.7351e-162
	-0.0515e-161	0.0290e-015	0	0.9880	0	-0.0002e-162
	-0.0560e-161	-0.0019e-015	0	0.9768	0	-0.2958e-162

	0.0199e-161	-0.0085e-015	0	0.9544	0	-0.0735e-162
	-0.0761e-161	0.0451e-015	0	0.9107	0	0.6420e-162
	0.0900e-161	-0.0605e-015	0	0.8294	0	0.8343e-162
	-0.4449e-162	-0.0127 e-015	-0.0487e-008	-0.9903	0	0.2178e-162
30	-0.9582e-162	0.0223 e-015	0.3553e-008	0.9939	0	0.1576e-162
	0.0920e-162	-0.0470 e-015	-0.1015e-008	0.9950	0	-0.0062e-162
	0.3226e-162	-0.0068 e-015	0.4905e-008	0.9954	0	-0.3472e-162
	0.3358e-162	-0.0420 e-015	-0.1912e-008	0.9946	0	-0.0341e-162
	0.3160e-162	0.0470 e-015	-0.1338e-008	0.9914	0	-0.0882e-162
	0.3977e-162	0.0030 e-015	0.0620e-008	0.9890	0	0.0415e-162
	-0.0174e-162	-0.0033 e-015	0.2949e-008	0.9962	0	-0.0552e-162
	-0.4467e-162	-0.1002 e-015	-0.2209e-008	0.9990	0	-0.1521e-162
	0.7180e-162	0.0170 e-015	-0.3675e-008	1.0009	0	-0.0591e-162
	-0.2169e-162	0.0248 e-015	0.0404e-008	0.9988	0	-0.3871e-162
	0.8562e-162	0.1148 e-015	0.1957e-008	0.9934	0	-0.0482e-162
	-0.2065e-162	-0.0830 e-015	-0.1759e-008	0.9849	0	0.1180e-162
	0.1602e-162	-0.0440 e-015	-0.7999e-008	0.9711	0	0.3318e-162
	-0.3398e-162	-0.0654 e-015	-0.3856e-008	0.9456	0	-0.0863e-162
	0.0887e-162	-0.0041 e-015	-0.5226e-008	0.8981	0	-0.3015e-162
	-0.3453e-162	0.0224 e-015	-0.0295e-008	0.8099	0	0.0456e-162
	-0.6019e-162	-0.0198 e-015	-0.0996e-008	0.6790	0	-0.1466e-162
	-0.0993e-162	-0.0324 e-015	-0.2310e-008	0.4819	0	0.0864e-162
	-0.0682e-162	-0.0693 e-015	0.2487e-008	0.2483	0	-0.4601e-162
	-0.1810e-162	-0.0647 e-015	0.0223e-008	0.0715	0	-0.0770e-162
	0.0210e-162	-0.0835 e-015	0.0664e-008	0.0138	0	0.3984e-162
	-0.6812e-162	-0.0414 e-015	0.1238e-008	0.0162	0	-0.2735e-162
	0.1254e-162	0.0051 e-015	-0.3041e-008	0.0108	0	-0.0137e-162
	-0.2725e-162	-0.0442 e-015	-0.1454e-008	0.0093	0	-0.1695e-162
	-0.2562e-162	0.0094 e-015	0.7564e-008	0.0108	0	-0.0944e-162
	-0.9082e-162	0.0061 e-015	0.0767e-008	0.0120	0	-0.1033e-162
	-0.4568e-162	0.0534 e-015	0.0545e-008	0.0181	0	-0.0107e-162
	-0.2781e-162	0.0661 e-015	0.1611e-008	0.0099	0	0.2920e-162
	-0.0782e-162	-0.0332 e-015	0.5492e-008	-0.0027	0	-0.3155e-162

Appendix-I

The MATLAB coding for the BF-PSO has given below which implemented in MATLAB 2009a.

######### Example_Functions.m file ###########

```
%#### Write the several benchmark functions for BF-PSO here example of
sphere is given ###
function fitns_fn=Fitness_fn(x) % Sphere function
fitns_fn=sum(x.^2);
end
```

########### Function_limit.m file #######

```
%#### Give the limits of benchmark functions for BF-PSO ###

function[minm,maxm]=Function_limit(Example_Function)

    minm=-100; % minimum value of dmension (or variables)

    maxm=100; % maximum value of dmension (or variables)

end
```

########### BF-PSO.m file #############

```
%####### Butterfly-PSO (or BF-PSO)#########
clear all
clc
format short
global bf
global dm
global Example_Function
[minm,maxm]=Function_limit(Example_Function);
ft_fn='Fitness_fn';
%############### Initialization ##############
 flight_max  = 1000; % Max number of flights by BF(or iterations)
itera_max=flight_max;
bf = 20;    % Size of the butterfly swarm
dm = 30;    % dimensions, 2, 10, 20, 30 ( or problem variables)
c1 =2;  % Acceleration rate C1
c2 =2;  % Acceleration rate C2
pg=1;
locations = rand(dm, bf).*(maxm-minm)+minm;
%%%% velocity limits as per variable problems;%%%%%%
vmax=((maxm-minm)/(rand*maxm));
vmin=-vmax;
velocity = rand(dm, bf).*(vmax-vmin)+vmin;
current_fitness = feval(ft_fn,locations);
local_best_locations  = locations;
mean_best_locations=mean(local_best_locations,2);
local_best_fitness  = current_fitness;
[global_best_fitness,Ind] = min(local_best_fitness);
globl_best_locations = local_best_locations(:,Ind);
     %%%%%%%%%%% Main iterations counter Loop %%%%%%%%%
```

```
itera=0;

while (itera < itera_max) % start main iteration loop

      itera = itera+1;
p=ones(dm,bf)*(global_best_fitness./(sum(local_best_fitness)));  %%%
probability of nectar

        ptt=(1-pt);

s=ones(dm, bf)*exp(-(itera_max-itera)/itera_max); % sensitivity of
butterflies

w=(itera_max-itera)/itera_max;

velocity=w.*velocity+ptt.*s.*c1.*rand(dm,bf).*(local_best_locations-
locations)+...

pg.*c2.*rand(dm,bf).*((globl_best_locations-locations)*ones(1,bf));   %
velocity updation

velocity=min(vmax,max(vmin,velocity));% check velocity limits

locations=locations+rand.*pt.*velocity);

% location updation

locations=min(maxm, max(minm,locations)); % check location or variable
limits

%%%%%%%%%%%%%%%%%% Updating local, mean and global best %%%%%%%

current_fitness=feval(ft_fn,locations);

Ifit=find(current_fitness<local_best_fitness);

local_best_fitness(:,Ifit)=current_fitness(:,Ifit);

local_best_locations(:,Ifit)=locations(:,Ifit);

 [current_global_best_fitness,Ind]=min(local_best_fitness);

    if current_global_best_fitness < global_best_fitness

       global_best_fitness = current_global_best_fitness;

    end

globl_best_locations=local_best_locations(:,Ind);

minfit2(itera+1)=global_best_fitness;

% record min fitness value for plot

end % end of main iteration loop

min_fit=global_best_fitness

globl_best_locati=globl_best_locations
```

```
bf_mean_fit=mean(minfit2)
bf_stdvn_fit=std(minfit2)

%%%%%% plots %%%%%%%
itera1=0:length(minfit2)-1;
plot(itera1, minfit2,'b')
xlabel('Iterations');
ylabel('Fitness Value');
legend('BF-PSO')
```