I.J. Intelligent Systems and Applications, 2018, 6, 59-67
Published Online June 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2018.06.07

Performance Estimation of Differential Evolution, Particle Swarm Optimization and Cuckoo Search Algorithms

Pankaj P. Prajapati
EC Department, L. D. College of Engineering, Ahmedabad, 380001, India
E-mail: pankaj@ldce.ac.in

Dr. Mihir V. Shah
EC Department, L. D. College of Engineering, Ahmedabad, 380001, India
E-mail: mihirec@gmail.com

Received: 30 April 2017; Accepted: 15 September 2017; Published: 08 June 2018

Abstract**—**Most design optimization problems in engineering are in general extremely nonlinear and deal with various design variables under complex restrictions. Traditional mathematical optimization procedure may fail to find the optimum solution to real-world problems. Evolutionary Algorithms (EAs) can serve as an efficient approach for these types of optimization problems. In this paper, Particle Swarm Optimization (PSO), Differential Evolution (DE) and Cuckoo Search (CS) algorithms are used to find the optimal solution for some typical unimodal and multimodal benchmark functions. The source codes of all these algorithms are developed using C language and tested on a core i5, 2.4 GHz processor with 8 GB internal RAM. PSO algorithm has a simplicity of implementation and good convergence speed. In contrast, CS algorithm has good ability to find a global optimum solution. To use the advantages of CS and PSO algorithms, a hybrid algorithm of CS and PSO (CSPSO) is implemented and tested with the same benchmark functions. The experimental simulation results obtained by all these algorithms show that hybrid CSPSO outperforms with PSO, DE and CS algorithms**.**

Index Terms**—**Optimization, Benchmark Function, Unimodal, Multimodal, Differential Evolution Algorithm, Particle Swarm Optimization Algorithm, Cuckoo Search Algorithm, Hybrid Algorithm.

I. Introduction

Optimization is a branch of science in which the best values of parameters of the problems are discovered up to certain limitations. There are various optimization procedures available in the literature. Optimization problems can be classified like Linear programming, Integer programming, Combinatorial optimization and Metaheuristic optimization [1] [2]. Linear programming and Integer programming methods could not find global optimum solutions for a non-deterministic polynomial time (NP)-hard problems with large numbers of variables and non-linear objective functions [3]. Metaheuristic algorithms are stochastic search procedures which mimic the natural biological advancement and/or the social manners. Metaheuristic optimization algorithms find optimum solution of an objective function using global exploration and local exploitation. The performance of various optimization algorithms can be estimated using various unimodal and multimodal benchmark functions. Many metaheuristic optimization algorithms have been developed in last few years. As per No Free Lunch (NFL) theorem, no single algorithm is best suited to solve all optimization problems [4]. So this paper depicts the hybrid algorithm concept for maximizing the performance by collaborating different metaheuristic algorithm.

II. Related works

Genetic Algorithm (GA) developed by Holland is the most popular and oldest optimization algorithm [5] [6]. Reference [7] describes basics of Particle Swarm Optimization (PSO) algorithm proposed by J. Kennedy and R. Eberhart. PSO algorithm has good local exploitation. Ant Colony Optimization (ACO) algorithm has been inspired by the foraging behavior of real ants [8]. The Differential Evolution (DE) algorithm developed by Kenneth Price and Rainer Storn is a population-based stochastic method for global optimization [9]. Artificial Bee Colony (ABC) algorithm proposed by Dervis Karaboga is motivated by the intelligent behavior of honey bees [10]. Cuckoo Search (CS) algorithm is the latest metaheuristic algorithm proposed by Yang and Deb [11]. The CS algorithm has a better explorative capability. Metaheuristic algorithms developed in recent years by many researchers are Firefly Algorithm (FA) [12] [13], Bat algorithm [14] [15], Stochastic Fractal Search (SFS) [16], Wind Driven Algorithm (WDA) [17], Grey Wolf Optimizer (GWO) [18], Symbiotic Organisms Search

(SOS) [19] and Grass Fibrous Root Optimization Algorithm (GRA) [20]. A comparison of GA, ACO, Reverse Formation Dynamics (RFD), FA and CS algorithm for Traveling Sales Man problem had been studied in [21]. In this work, the performance of PSO, DE, CS and hybrid algorithm of CS and PSO (CSPSO) is evaluated using 11 standard benchmark functions.

The content of this paper is structured as follows. The PSO, DE and CS algorithms are briefly described in section III. The concept of hybrid CSPSO algorithm is represented in section IV. The various standard benchmark functions used for the testing purpose are presented in section V. Experimental settings and simulation results are presented in section VI. Conclusions are discussed in section VII.

III. DE, PSO AND CS ALGORITHMS

A. Differential Evolution Algorithm

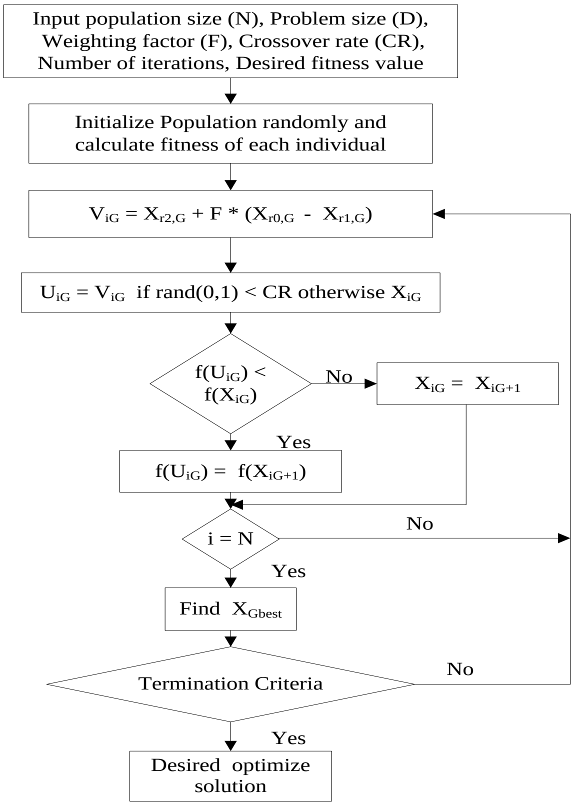
The DE algorithm uses the mutation, crossover, and selection strategies of GA. There are many mutation techniques for DE algorithm given in literature [9] [22]. The DE/rand/1/bin is the most commonly used technique to search the global optimum [22]. DE algorithm consist of two arrays, each one holding population size N and dimension D. The first array holds the current vector population, while the second array holds vector population for the next generation. In each generation, N competitions are carried out to get the resultant composition of the consecutive generation. The vector differential ($X_{r0} – X_{r1}$) is defined by a pair of vectors X_{r0} and X_{r1}. Both vectors X_{r0} and X_{r1} are chosen at random, and their weighted difference is multiplied by some weighted factor which is further added to another randomly chosen vector X_{r2}. This process can be mathematically represented using following equation.

$$X = X_{r2} + F * (X_{r0} - X_{r1}) \tag{1}$$

The weighting factor F is a constant supplied by the user in the optimal range of (0.5, 1) [22]. In DE algorithm main control parameters are F, N and crossover rate (CR) [22]. The main steps of DE algorithm are given by the flow diagram in Fig. 1.

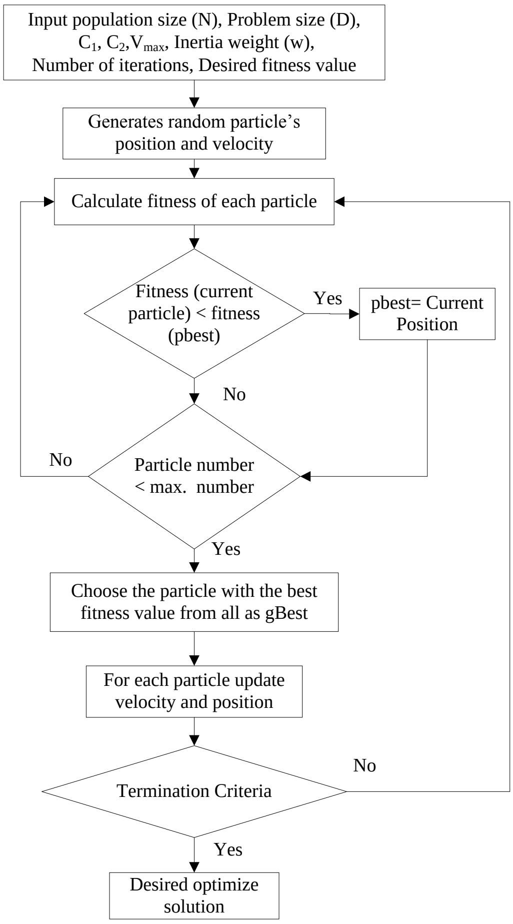
B. Particle Swarm Optimization Algorithm

The PSO algorithm is basically a swarm intelligent based evolutionary algorithm [7]. It finds the best optimal result using a set of flying birds with different velocities. The velocities of these birds are dynamically adjusted based on their past performance, as well as their neighbor in the exploration space. In this algorithm, each solution bird in folk is known as a particle. The birds in the population change their social performance based on their movement toward the destination. This algorithm works in an iteration manner and finds the best solution. The main steps for PSO algorithm are given by the flow diagram as shown in Fig. 2.



Input population size (N), Problem size (D), Weighting factor (F), Crossover rate (CR), Number of iterations, Desired fitness value
Initialize Population randomly and calculate fitness of each individual
$V_{iG} = X_{r2,G} + F * (X_{r0,G} - X_{r1,G})$
$U_{iG} = V_{iG}$ if rand(0,1) < CR otherwise X_{iG}
$f(U_{iG}) < f(X_{iG})$
No
$X_{iG} = X_{iG+1}$
Yes
$f(U_{iG}) = f(X_{iG+1})$
i = N
No
Yes
Find X_{Gbest}
Termination Criteria
No
Yes
Desired optimize solution

Fig.1. Flowchart of DE Algorithm.



Input population size (N), Problem size (D), C_1, C_2, V_{max}, Inertia weight (w), Number of iterations, Desired fitness value
Generates random particle's position and velocity
Calculate fitness of each particle
Fitness (current particle) < fitness (pbest)
Yes
pbest= Current Position
No
No
Particle number < max. number
Yes
Choose the particle with the best fitness value from all as gBest
For each particle update velocity and position
Termination Criteria
No
Yes
Desired optimize solution

Fig.2. Flowchart of PSO Algorithm

Suppose particle swam consists of N number of particles with D dimension. The position and velocity of the kth particle are characterized by $X_k = [X_k^1, X_k^2, . . .,X_k^D]$ and $V_k = [V_k^1, V_k^2,. . .,V_k^D]$. The velocity of the kth particle after each iteration is updated according to the equation given below [13].

$$\begin{aligned} V_{kd}^{i+1} &= W * V_{kd}^{i} + C_1 * rand_1 * \left(pbest_{kd}^{i} - X_{kd}^{i} \right) \\ &+ C_2 * rand_2 * \left(gbest_{d}^{i} - X_{kd}^{i} \right) \end{aligned} \quad (2)$$

Here, the range of k is {1, 2, 3........., N}, the range of d is {1, 2, 3…., D}, the range of i is {1, 2, 3.... maximum iteration number}, pbest is particle's personal current best position, gbest is particle's global best position, $rand_1$ and $rand_2$ are uniformly distributed random numbers between 0 and 1, C_1 and C_2 are constants known as acceleration coefficients, W is known as inertia weight which is initially chosen less than one and then reduced linearly with each iteration. W controls the influence of the previous direction of displacement. The position of the kth particle in N x D dimension of the search space can be calculated using the following equation.

$$X_{kD}^{i+1} = X_{kD}^{i} + V_{kD}^{i+1} \quad (3)$$

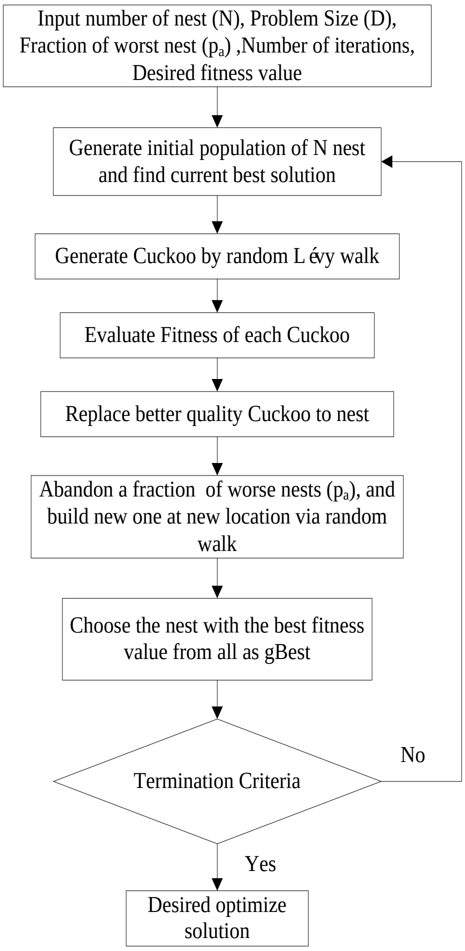

Input number of nest (N), Problem Size (D), Fraction of worst nest (p_a) ,Number of iterations, Desired fitness value
Generate initial population of N nest and find current best solution
Generate Cuckoo by random Lévy walk
Evaluate Fitness of each Cuckoo
Replace better quality Cuckoo to nest
Abandon a fraction of worse nests (p_a), and build new one at new location via random walk
Choose the nest with the best fitness value from all as gBest
Termination Criteria
No
Yes
Desired optimize solution

Fig.3. Flowchart of CS Algorithm

C. Cuckoo Search Algorithm

The CS algorithm developed by Xin-She Yang and Suash Deb is a population-based stochastic global search algorithm [11]. This algorithm is motivated by the exceptional lifestyle of the cuckoo species. The main steps of CS algorithm are given by the flow diagram in Fig. 3. In this algorithm, each pattern relates to a nest and each individual element of the pattern relates to a Cuckoo egg [11]. The common process equation of the CS algorithm is represented by following equation [11].

$$X_{g+1;i} = X_{g;1} + \alpha \otimes L\acute{e}vy(\lambda) \quad (4)$$

Here, α > 0 is the step size which depends on scales of the problem of interests and g represents the number of the current generation. The product ⊗ indicates the entrywise multiplication. Lévy (λ) is random walk using Lévy flight which is more effective in the long run as compared to a random walk in PSO algorithm. Lévy (λ) is derived from a Lévy distribution with an infinite variance and infinite mean [11].

$$L\acute{e}vy \sim \mathrm{u} = t^{-\lambda} \quad (5)$$

Here, step size follows random walk process with a power law distribution with heavy-tailed. CS is a population-based stochastic optimization algorithm similar to DE and PSO algorithms, but it uses some sort of elitism and/or selection similar to that used in harmony search [11]. The randomization is more efficient in CS algorithm as compared to PSO and DE algorithms [11]. CS algorithm requires fewer parameters to be adjusted as compared to DE and PSO algorithms [11].

IV. Hybrid CSPSO Algorithm

In this section, a hybrid algorithm of CS and PSO algorithms is presented. The main steps of the hybrid CSPSO algorithm are given by the flow diagram in Fig. 4. PSO is one of the most capable optimization algorithms, but it converges rapidly so it has premature convergence generally in complex problems [23].

The CS algorithm is used to solve a number of complex problems and outperformed other algorithms [24]. CS algorithm may converge marginally slower but it has better explorative skill. There is no algorithm that can efficiently optimize all optimization problems. To resolve this issue, merging of the current algorithms can be done to find the global optimum solution [25]. The convergence rate and global search ability can be improved using the hybrid algorithm. It can decrease the possibility of being stuck in local minima. The hybrid algorithm can be designed to make the best use of best features of both algorithms.

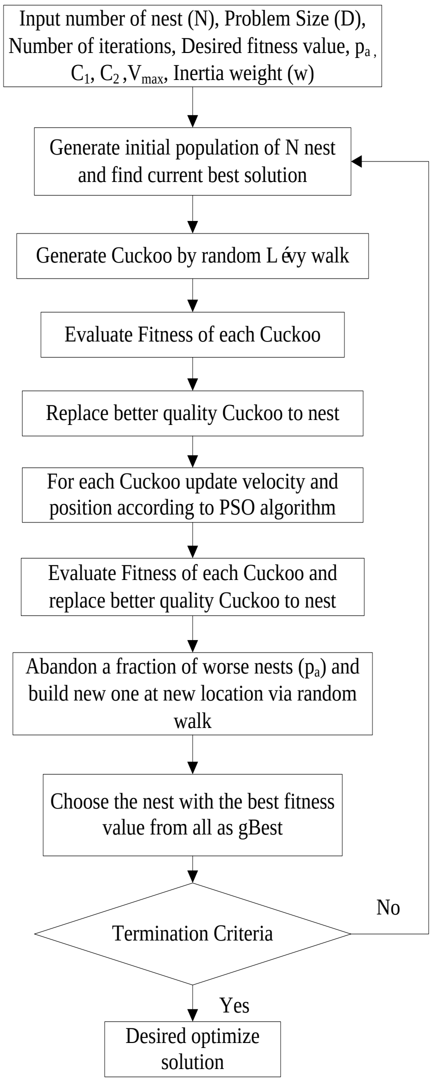

Input number of nest (N), Problem Size (D), Number of iterations, Desired fitness value, p_a, C_1, C_2, V_{max}, Inertia weight (w)
Generate initial population of N nest and find current best solution
Generate Cuckoo by random L évy walk
Evaluate Fitness of each Cuckoo
Replace better quality Cuckoo to nest
For each Cuckoo update velocity and position according to PSO algorithm
Evaluate Fitness of each Cuckoo and replace better quality Cuckoo to nest
Abandon a fraction of worse nests (p_a) and build new one at new location via random walk
Choose the nest with the best fitness value from all as gBest
Termination Criteria
No
Yes
Desired optimize solution

Fig.4. Flowchart of CSPSO Algorithm

V. BENCHMRK FUNCTIONS

Performance of an optimization algorithm can be estimated using typical standard benchmark functions [26]. There are different types of such test functions are represented in the literature. Some of the benchmark functions we have been used for performance evaluation of DE, PSO, CS and hybrid CSPSO algorithms. The benchmark functions used in this work are listed in Table 1 with their search space, dimension and desired fitness value.

VI. EXPEREMINTALSETTINGS AND RESULTS

All these algorithms were implemented using C language and tested on a system consisting of core i5, 2.4 GHz processor with 8 GB internal RAM and Ubuntu as an operating system. Every algorithm has its own parameters which vary its performance in terms of solution quality and processing time.

In DE algorithm, control variables setting of F, CR and N can be quite difficult and some benchmark functions are very sensitive to proper settings these control variables [22]. For this work, N = 10*D, F = 0.8 and CR = 0.5 are chosen for DE algorithm [22]. In PSO algorithm, we have used N = 30, C_1 = 1.47, C_2 = 1.47, V_{max} = X_{max}, V_{min} = X_{min} and W = [0.4, 0.9] [27]. CS algorithm requires less parameters as compared to DE and PSO algorithms. In CS algorithm, N = 30 and p_a = 0.25 are chosen [11]. In hybrid CSPSO algorithm, N = 30, C_1 = 1.47, C_2 = 1.47, V_{max} = X_{max}, V_{min} = X_{min}, W = [0.4, 0.9] and p_a = 0.25 are chosen.

The convergence graphs optimized by DE, PSO, CS and CSPSO algorithms of different benchmark functions are listed in Table 1 as shown in Fig. 5 to Fig. 15. The vertical axis is fitness value and the horizontal axis is the number of iterations. From Fig. 5 to Fig. 15, we can observe that the CS algorithm requires fewer iterations as compared to those in PSO algorithm to reach required fitness values for all benchmark functions. The CS algorithm also requires fewer iterations as compared to those in DE algorithm to reach required fitness values for all benchmark functions except in Rastrigin function. The convergence graphs also show that CSPSO algorithm outperforms DE, PSO and CS algorithms for almost all benchmark functions.

Table 1. Standard Benchmark Functions

Function	Dimension D	Type	Search Space	Fitness Value
Ackley	8	Multimodal	(-32, 32)	10^{-6}
Beale	2	Unimodal	(-4.5, 4.5)	10^{-6}
Bukin	2	Multimodal	-15<< X_1 < -5 -3 < X_2 < 3	10^{-6}
Easom	2	Unimodal	(-100, 100)	-1
Griewank	8	Multimodal	(-600, 600)	10^{-6}
L évy	8	Multimodal	(-10, 10)	10^{-6}
Michalewicz	10	Multimodal	(0, π)	-9.66
Rastrigrin	8	Multimodal	(-5.12, 5.12)	10^{-6}
Rosenbrock	8	Unimodal	(-30, 30)	10^{-6}
Schaffer	2	Multimodal	(-100, 100)	10^{-6}
Sphere	8	Unimodal	(-100, 100)	10^{-6}

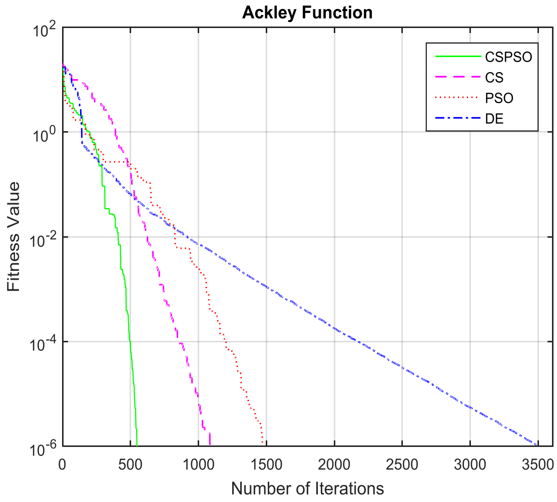


Fig.5. Convergence graph of DE, PSO, CS and CSPSO algorithms for Ackley function

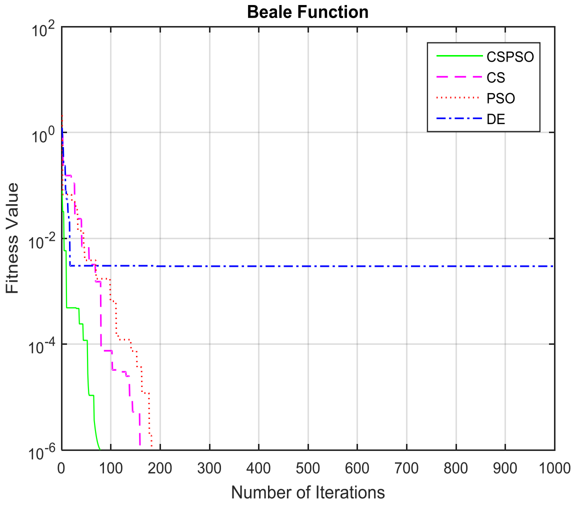


Fig.6. Convergence graph of DE, PSO, CS and CSPSO algorithms for Beale function

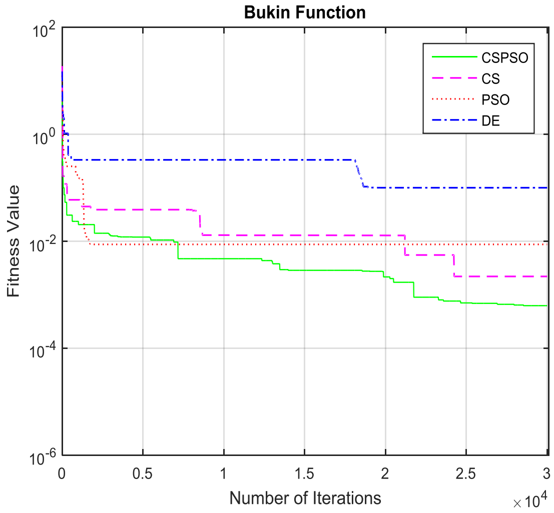


Fig.7. Convergence graph of DE, PSO, CS and CSPSO algorithms for Bukin function

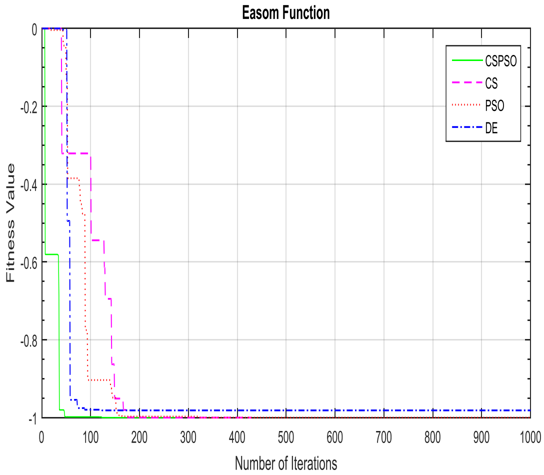


Fig.8. Convergence graph of DE, PSO, CS and CSPSO algorithms for Easom function

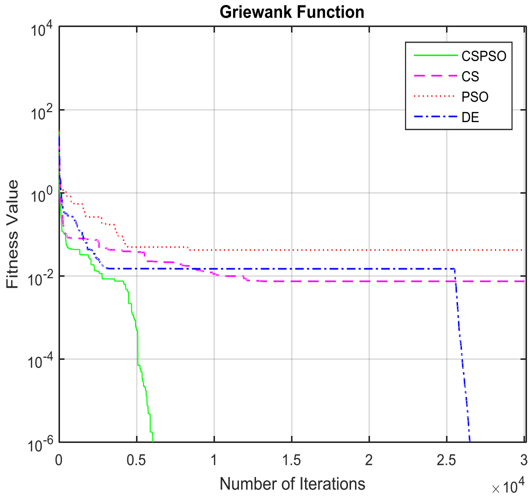


Fig.9. Convergence graph of DE, PSO, CS and CSPSO algorithms for Griewank function

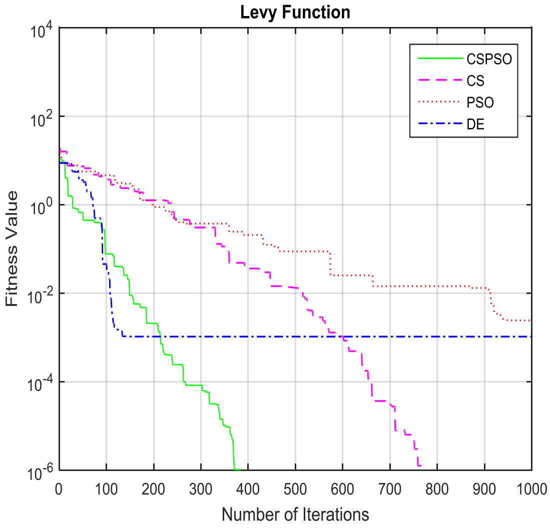


Fig.10. Convergence graph of DE, PSO, CS and CSPSO algorithms for L évy function

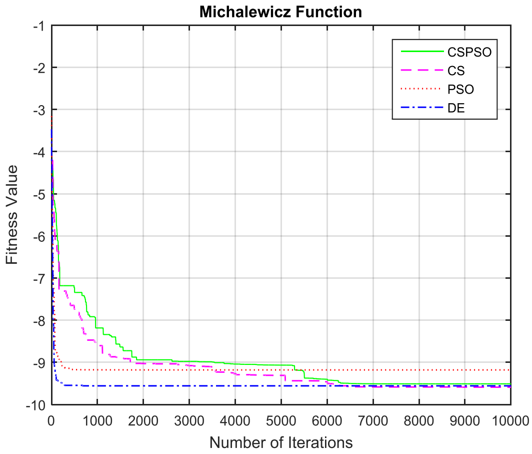


Fig.11. Convergence graph of DE, PSO, CS and CSPSO algorithms for Michalewicz function

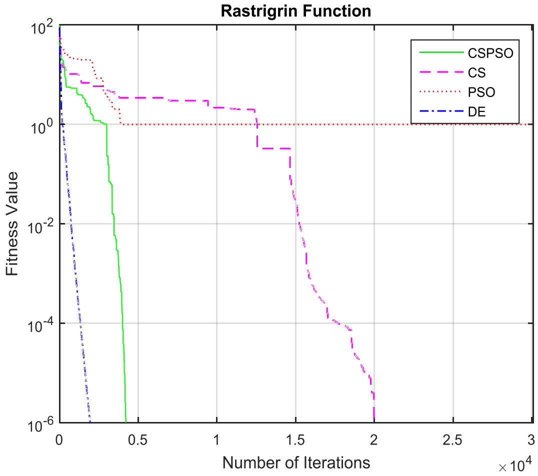


Fig.12. Convergence graph of DE, PSO, CS and CSPSO algorithms for Rastrigrin function

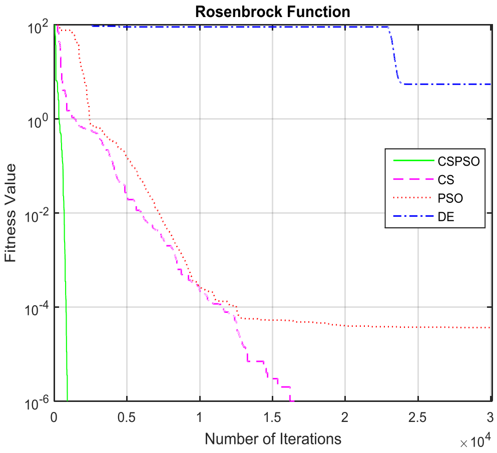


Fig.13. Convergence graph of DE, PSO, CS and CSPSO algorithms for Rosenbrock function

Each experiment is repeated for 100 independent runs with different random seeds. The standard deviation, average number of iterations and simulation time obtained by DE, PSO, CS and CSPSO algorithms are given in Table 2. From Table 2, it is observed that PSO algorithm requires less simulation time for all functions. CS and CSPSO require fewer iterations to reach required fitness value. CS and CSPSO give less standard deviation for all functions.

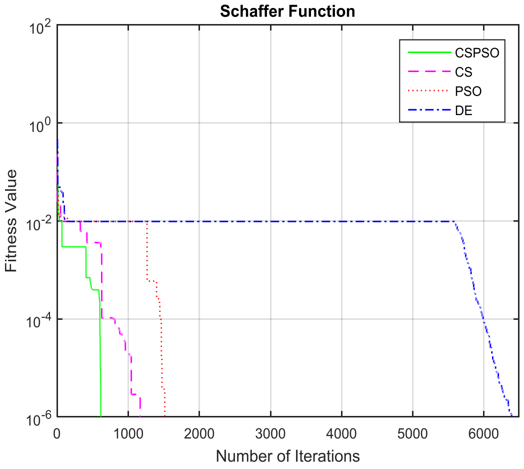


Fig.14. Convergence graph of DE, PSO, CS and CSPSO algorithms for Schaffer function

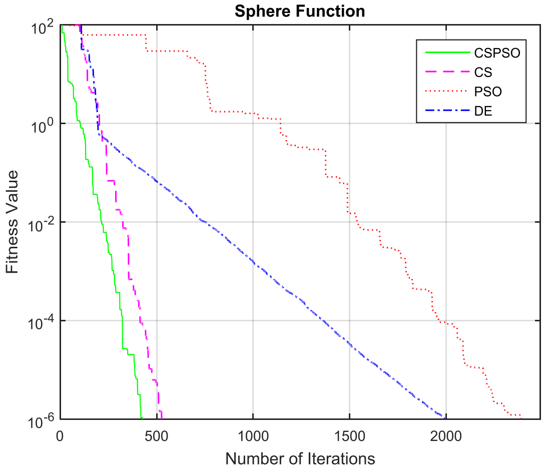


Fig.15. Convergence graph of DE, PSO, CS and CSPSO algorithms for Sphere function

VII. Conclusion

PSO, DE and CS algorithms are implemented using C language and tested on 11 unimodal and multimodal nonlinear benchmark functions. CS algorithm requires fewer control parameters as compared to PSO and DE algorithms. Experimental simulation results show that CS algorithm is more efficient as compared to PSO algorithm for all unimodal and multimodal functions. The CS algorithm also outperforms DE algorithm except for Rastrigrin function. PSO algorithm is most commonly used algorithm due to its easiness of implementation and fast convergence speed. In contrast, CS algorithm has good ability to find a global optimum solution. To use the advantages of CS and PSO algorithms, a hybrid algorithm of CS and PSO is implemented and tested with the same benchmark functions. The simulation results show that hybrid CSPSO algorithm outperforms PSO, DE and CS algorithms.

Table 2. Simulation Results

Algorithm	Function	Standard Deviation	Average Iterations	Simulation Time (s)
DE	Ackley	0.0	4021	28.69
PSO		0.0	1521	5.60
CS		0.0	1107	16.59
CSPSO		0.0	553	10.53
DE	Beale	0.0046	28514	42.75
PSO		0.0	580	0.83
CS		0.0	154	0.60
CSPSO		0.0	89	0.42
DE	Bukin	0.0048	30000	37.29
PSO		0.0122	30000	22.92
CS		0.0032	30000	119.20
CSPSO		0.0035	30000	137.79
DE	Easom	0.0069	30000	59.49
PSO		0.0	1051	1.45
CS		0.0	480	2.54
CSPSO		0.0	311	2.09
DE	Griewank	0.0063	20529	168,69
PSO		0.0170	29514	111.33
CS		0.0124	26530	408.61
CSPSO		0.0123	25071	471.54
DE	Lévy	0.0082	30000	294.62
PSO		0.0	2147	10.01
CS		0.0	746	12.11
CSPSO		0.0	387	8.27
DE	Michalewicz	0.1065	22261	333.86
PSO		0.4183	30000	159.79
CS		0.1163	26166	544.96
CSPSO		0.1267	26122	719.06
DE	Rastrigrin	0.0	17776	92.05
PSO		0.3233	7073	20.72
CS		0.532	10550	146.47
CSPSO		0.4532	8805	157.10
DE	Rosenbrock	19.39	30000	114.56
PSO		8954.35	29279	82.19
CS		0.0	1234	14.52
CSPSO		0.0	993	13.66
DE	Schaffer	0.0	6245	4.71
PSO		0.0	786	0.83
CS		0.0	1258	5.89
CSPSO		0.0	462	2.65
DE	Sphere	0.0	1904	6.78
PSO		0.0	2662	4.95
CS		0.0	568	6.87
CSPSO		0.0	416	5.59

ACKNOWLEDGMENT

I wish to thank my guide Dr. Mihir V. Shah and God for inspiring and supporting me for this effort because without their guidance this work would not have been possible. Finally, I passionately thank my parents, my wife and my lovely family for their kind support and patience.

REFERENCES

[1] M. Dorigo and G. Di. Caro, New ideas in optimization, *McGraw-Hill Ltd.*, UK, ISBN: 0-07-709506-5, 1999.

[2] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multi-objective Genetic Algorithm: NSGA-II, *IEEE Transaction on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002. DOI:10.1109/4235.996017

[3] M. Lovbjerg, Improving PSO by hybridization of stochastic search heuristics and self-organized criticality, *Master Thesis, Aarhus Universitet, Denmark,* 2002.

[4] D. H. Wolpert and W. G. Macread, No free lunch theorems for optimization, IEEE *Trans. On Evol. Comput.* Vol. 1, No. 1, pp. 67–82, 1997. DOI :10.1109/4235.585893

[5] J. Holland, Adaption in natural and artificial systems, *Ann Arbor, University of Michigan Press,* 1975.

[6] J. Naomi Rosenfield Boeira, The effects of "Preferentialism" on a Genetic Algorithm population over elitism and regular development in a Binary F6 fitness function, *International Journal of Intelligent Systems and Applications (IJISA),* Vol.8, No.9, pp.38-46, 2016. DOI: 10.5815/ijisa.2016.09.05

[7] J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, *Proc. of IEEE Int. Conf. on Neural Networks,* Piscataway, NJ, pp. 1942-1948, 1995.

[8] M. Dorigo, V. Maniezzo and A. Golomi, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybernet, Vol.* 26, No. 1, pp. 29–41, 1996. DOI: 10.1109/3477.484436

[9] R. Stron and K. Price, Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization,* Vol. 11, No. 4, pp. 341–359, 1997.

[10] D. Karaboga, An idea based on Honey Bee swarm for numerical optimization., *Technical Report-TR-06, Erciycs University,* Engineering Faculty, Computer Engineering Dept., 2005.

[11] X. S. Yang and S. Deb, Cuckoo search via L ́evy flights, *in Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), India. IEEE Publications, USA,* pp. 210-214, 2009.

[12] X. S. Yang, Firefly algorithm, stochastic test functions and design optimization, *International Journal of Bio-Inspired Computation,* Vol. 2, No. 2, pp.78–84, 2010.

[13] Yongbo Sui, Lingzhi Yi and Wenxin Yu, A novel and improved Firefly Algorithm based on two order oscillation, *International Journal of Intelligent Systems and Applications (IJISA),* Vol.9, No.5, pp.19-26, 2017. DOI: 10.5815/ijisa.2017.05.03

[14] X. S. Yang, A new metaheuristic Bat-inspired Algorithm, *in J. R. Gonzalez et al. (Eds.): Nature Inspired Cooperative Strategies for Optimisation (NICSO 2010),* Vol. 284, pp.65–74, Springer, SCI, 2010.

[15] Menga, X. Z. Gaob, Y. Liuc and H. Zhanga, A novel Bat Algorithm with habitat selection and Doppler effect in echoes for optimization, *Expert Systems with Applications,* Vol. 42, No. 17–18, 6350–6364, 2015.

[16] H. Salimi, Stochastic Fractal Search: A Powerful Metaheuristic Algorithm, *Knowledge-Based Systems,* Vol. 75, pp. 1-18, 2015.

[17] Z. Bayraktar, M. Komurcu and D. H. Werner, Wind Driven Optimization (WDO): A novel Nature Inspired Optimization Algorithm and its application to electromagnetics, *IEEE Int. conf., Antennas and Propagation Society International Symposium (APSURSI), Toronto,* pp. 1-4, 2010.

[18] S. Mirjalilia, S. M. Mirjalilib and A. Lewisa, Grey Wolf Optimizer, *Advances in Engineering Software,* Vol. 69, pp. 46-61, 2014. DOI: http://dx.doi.org /10.1016/j.advengsoft. 2013.12.007

[19] M. Cheng and D. Prayogo, Symbiotic Organisms Search: A new metaheuristic optimization algorithm, *Computers & Structures,* Vol. 139, pp. 98–112, 2014.

[20] Hanan A. R. Akkar and Firas R. Mahdi, Grass Fibrous Root Optimization Algorithm, *International Journal of Intelligent Systems and Applications (IJISA),* Vol.9, No.6, pp.15-23, 2017. DOI: 10.5815/ijisa.2017.06.02

[21] B. V. Chawda and J. M. Patel, Investigating performance of various natural computing algorithms, *International Journal of Intelligent Systems and Applications (IJISA),* Vol. 9, No.1, pp.46-59, 2017. DOI: 10.5815/ijisa. 2017.01.05

[22] V. Arunachalam, Optimization using Differential Evolution, *Water Resources Research Report no. 60, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada,* July 2008.

[23] P. Civicioglu and E. Besdok, A conceptual comparison of the Cuckoo Search, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony algorithms, *Springer, Science+Business Media B.V.,* July 2011.

[24] I. Fister Jr., D. Fister and I. Fister, A comprehensive review of Cuckoo Search: variants and hybrids, *International Journal of. Mathematical Modelling and Numerical Optimization,* Vol. 4, No. 4, 2013.

[25] Roy and S. Chaudhuri, Cuckoo Search Algorithm using Lѐvy flight: A Review, *International Journal of Modern Education and Computer Science,* Vol. 5, No.12, pp. 10-15, Dec-2013. DOI: 10.5815/ijmecs.2013.12.02

[26] M. Jamil and X. S. Yang, A literature survey of benchmark functions for global optimization problems, *Int. Journal of Mathematical Modeling and Numerical Optimization,* Vol. 4, No. 2, pp. 150–194, 2013.

[27] R. A. Thakker, M. Shojaei Baghini and M. B. Patil, Low-power low-voltage analog circuit design using Hierarchical Particle Swarm Optimization, *IEEE VLSI Design,* pp. 427-432, 2009.

Authors' Profiles

Pankaj P. Prajapati is currently working as an assistant professor in Electronics and Communication (EC) Department at L. D. College of Engineering, Gujarat, India. He has received B.E. in EC Engineering from L. D. College of Engineering in 2001 and M.Tech. in EC with specialization in VLSI Design from Institute of Technology, Nirma University in 2009. He has completed his dissertation work at ISRO, Ahmedabad. He has 15 years of teaching experience at various colleges. He is currently pursuing Ph.D. in the area of optimization of CMOS based analog circuits from Gujarat Technological University (GTU). He has published many research papers in International /National journals/conferences. He is the co-author of a book on VLSI Technology. He is a life member of ISTE and IETE. Currently, he is also serving as a local executive committee (LEC) member of IETE Ahmedabad Center.

Dr. Mihir V. Shah is working as a professor in EC Department at L. D. College of Engineering, Gujarat, India. He has completed B.E. in Electronics, M.E. from MNIT, Jaipur and Ph.D. from M.S. University, Baroda. His research interests are VLSI Design and Embedded System. He has total 25 years of experience in teaching and industry. He has published several papers in various International/ national journals/conferences. He is a life member of IETE. He has also served as a secretary for IETE Ahmedabad Centre and an expert committee member at GTU, Gujarat University, Commissionerate of Technical Education and in many private universities.

How to cite this paper: Pankaj P. Prajapati, Mihir V. Shah, "Performance Estimation of Differential Evolution, Particle Swarm Optimization and Cuckoo Search Algorithms", International Journal of Intelligent Systems and Applications(IJISA), Vol.10, No.6, pp.59-67, 2018. DOI: 10.5815/ijisa.2018.06.07