Cross Opposition Based Differential Evolution Optimization

Shweta Sharma, Ashwani Kumar Yadav, Deepak Sinwar, Bhagyashri Naruka, Vaishali

Abstract: Differential Evolutionary (DE) Algorithms is one of the most popular metaheuristic approach. For optimization purpose DE is very useful to solve various kind of problems. In addition to that the paper offers a Cross-Opposition Based Differential Evolution (CODE). An impression of Opposition-based learning (OBL) is incorporated in population initialization phase and in step of crossover. The performance of algorithm is analysed for different mutation strategies of DE and various other existing approaches. Results demonstrated that the algorithm outperform in terms of convergence speed, versatile population and dimension size.

Index Terms: Differential Evolutionary (DE), Cross-Opposition Based Differential Evolution (CODE), Opposition-based learning (OBL), Metaheuristic algorithm, Optimization

I. INTRODUCTION

Now a days Optimization is the key factor of all nature inspired algorithm. In a modest opinion Optimization can be defined as indicating the greatest substitute from the specified set of resolutions. Optimization is needed at each stage of human actions and different techniques of optimization is broadly used wherever conclusions are consumed in some or extra complex circumstances that contain mathematical formulation. For that purpose, stochastic search techniques are occupied to crack such intricate high dimension plus real-world optimization glitches.Stochastic optimization practices are directly related to nature inspired metaheuristic algorithms where algorithm takes their encouragement from natural surroundings or some living spectacle.

One of the most widespread algorithms is Differential Evolution [1]. Differential evolution algorithm becomes a trend setter of metaheuristic algorithm to solve different kind of problems like engineering problems, discrete and continuous problems, multimodal problems etc. and Opposition Based Learning (OBL) and combines the best features of them for better efficiency. The main idea It can be done due to the natural behaviour and intelligence of the algorithm. It works on the concept of 'survival of fittest'.The most successful variant of DE is Opposition Based Differential Evolution (OBDE) in 2006 [2]. OBDE incorporate the concept of Differential Evolution to hybridize these two approaches was to achieve best solution across global search space in large scale optimization also in complex problems. OBDE has also proves its great performance and capability in determining global optimal solutions towards numerous combinatorial optimization glitches [3]. The learning presents opposition-based crossover to boost the convergence speed of ODE. The resulting algorithm is named as Cross Opposition Based Differential Evolution (CODE).The paper is systematized as follows: Basic ODE is explained in Section 2, trailed by Section 3, which describes the proposed CODE and strategies used for simulation are given in Section 4. The Experiment Analysis and Parameterization is discussed in Section 5 trailed via discussion part in section 6. Finally, conclusion of learning is presented in Section 7.

II. OVERVIEW OF OPPOSITION BASED DIFFERENTIAL EVOLUTION

Opposition based Differential Evolution is combine approach of DE and OBL. These two techniques are hybridized because they have the capability of accelerate the convergence speed of evolutionary algorithms. As per any nature inspired algorithm, ODE follow the same steps like population is initialized on random bases. Then opposite population is generated by using the concept of opposition-based learning(OBL). After that it selects the elite population from original and opposite ones according to fitness function. Population now go for mutation crossover and selection process to produce new generation until the stopping criteria not met. The following steps of ODE are:

(i)Initial population: A random generation is selected to be optimized using the following equation

$$Pop=L+(H-L)*rand(0,1) \quad \ldots(1)$$

After generation of initial population, opposite generation is produced like

Opop (Pop) =

Revised Manuscript Received on 30 July 2019.
* Correspondence Author

Shweta Sharma*, Department of Computer Science & Engineering, Amity School of Engineering & Technology at Amity University Rajasthan, Jaipur, India.

Ashwani Kumar Yadav, Department of Electronics & Communication Engineering, Amity School of Engineering & Technology at Amity University Rajasthan, Jaipur, India.

Deepak Sinwar, Department of Computer and Communication Engineering, School of Computing & Information Technology at Manipal University Jaipur, Jaipur, Rajasthan, India.

Bhagyashri Naruka, Department of Computer Science & Engineering, Amity School of Engineering & Technology at Amity University Rajasthan, Jaipur, India.

Vaishali, Department of Computer and Communication Engineering, School of Computing & Information Technology at Manipal University Jaipur, Jaipur, Rajasthan, India.

Retrieval Number B2283078219/19©BEIESP
DOI: 10.35940/ijrte.B2283.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\min(Pop)+\max(Pop)-Pop \quad \ldots(2)$$

Using eq. (1) and (2) , elite population is selected to calculate trial vector.

(ii) Mutation: separately taking target vector of current population for generating a mutant vector as follow

$$Mutant = Pop\,(a)+F*(Pop(b)-Pop(c)) \quad \ldots(3)$$

where a, b, c are random indexes and F is the scaling factor

(iii) Crossover: to achieve the diversity of perturbation of mutant vector crossover takes place. So by the end of thos step trial vector is accomplished.

(iv) Selection: 'The survival of fittest' is actually applied at this stage. This is the deciding phase where value of fitness function of trial and target vector is compared. Selection is done on greedy bases. If the fitness of value of trial vector is higher then target vector then it is selected for further optimization otherwise old valve will be taken care of for next process.

III. CROSS OPPOSITION BASED DIFFERENTIAL EVOLUTION

All optimization algorithms follow the practically similar practice that is why they all are pretty same with respect to results obtain by them. The main step for optimization is population initialization and selection of offspring for the new generation. To reduce the computational cost and improve the convergence speed of algorithm we need to update the random production of population. These modifications are already being done in Differential Evolution (DE) from the decades by many researchers. The successful variant of these amendment is Opposition Based Differential Evolution (OBDE) introduced by Rahnamayan & Tizhoosh in [5-7]. The following are the main steps of ODE:

1) Population initialization,
2) Mutation,
3) Crossover and
4) Selection.

In recent years, ODE is enhanced in ways on different steps [14-20]. The proposed algorithm imposes OBL on crossover phase of ODE algorithm. The technique is adapted for better convergence speed and to solve complex problem in utilized manner. The pseudo code of proposed algorithm as follow:

```
Begin;
Generate initial random population Pop(D, NP)
        Calculate the fitness function f(Pop) and sort
        accordingly
Generate its random opposite population OPop(NP, D)
        Calculate the fitness function f(OPop) and sort
        accordingly
Select elite population NPopi,j from Pop and OPop on the
bases of fitness function f(Pop) and f(OPop) respectively
For VTR<= $10^{-6}$
        Generate mutant vector mutanti,j for each vector
        NPopi,j
        Apply crossover operation newi,j
         Calculate opposite crossover
         Calculate trial vector

End:
```

IV. MODEL STRATEGY

Basic concentration of the current learning is on equating superiority of the simulated results. For investigation of convergence speed, algorithm is evaluated on the bases of number of function evaluation. A slighter change in NFE earns higher convergence rapidity. A slighter value than the value-to-reach (VTR) is the stopping criterion formerly accomplishment of maximum number of function evaluation. The results are noted for 50 runs in terms of NFEs, average (mean) and the standard deviations. The proposed algorithm Cross Opposition based Differential Evolution is compared with DE (differential evolution) and Opposition based Differential Evolution (ODE). The projected algorithm CODE is executed in MATLAB. For the comparison point of view, convergence speeds are calculated in terms of acceleration rate (AR). It is defined as follow:

$$AR = NFE_{DE/ODE}/NFE_{CODE} \quad \ldots(4)$$

If the value or AR is greater than 1 then it means CODE is faster than compared algorithm with respect to convergence speed.

V. PARAMETER SETUP

The parameter settings of DE, ODE and CODE remain same and specified in TABLE I intended for impartial comparison. The results of proposed algorithm are evaluated for 15 benchmark problems. The segment relates the performance of Cross ODE with basic DE (DE/rand/1/bin) and basic ODE. Parameter Setting in all experiments that are conducted, are as follow in Table I.

In the direction of preserving a consistent plus fair assessment, for entire experimentations: 1) the constraint settings for all experiments stand identical; 2) for all shown experiments, the results of 50 independent run is taken as an average and last but not the least; 3) extra fitness evaluation is also counted in NFEs which are needed for calculating opposite points.

Table I. Testing System Parameterization

Sr. No	Parameters	Value
1	Population Size (Pop)	100
2	Dimension (D)	30
3	Crossover Probability Constant (Cr)	0.9
4	Scaling Factor (F)	0.5
5	Mutation strategy	DE/rand/1/bin
6	Value to Reach (VTR)	10^{-6}

Retrieval Number B2283078219/19©BEIESP
DOI: 10.35940/ijrte.B2283.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. RESULT AND DISCUSSION

The comparison process starts from parent algorithm DE and ODE with CODE with respect to convergence speed and robustness. This can be done by analysing the TABLE II where Number of function evaluation (NFEs), Mean value(M) and Standard Deviation (SD) is given for all three algorithms (DE, ODE, CODE). The recommended method works better on greater dimension and population size. The experiment for higher dimensions leads to the better results. The results of solving 20 benchmark functions are given in TABLE II. The boldfacing value in TABLE II for all functions indicates the best result of the NFEs for that particular function. Correspondingly, Acceleration Rate (AR) is calculated for usual success rate and also on 20 test functions that are indicated in last two column of the table.

The proposed algorithm overtakes DE and ODE on almost each function (best performance on 16 functions), whereas DE surpasses ODE on very few functions (F_8 and F_{16}) otherwise ODE also perform well but not better than CODE. The projected algorithm is fail to solve some functions (F_8, F_{12}, F_{15} and F_{16}) so it is difficult to conclude about the performance of any particular algorithm for such cases.

The results prove that the Cross-Opposition Based Differential Evolution (CODE) exploit the search space very fast. So, the proposed algorithm performed well for convergence in comparison to existing algorithms. By applying OBL in population initialization phase as discussion in ODE, the algorithm converges faster. To optimize it more, the concept of OBL is applied to crossover operation. Most of the evolutionary algorithm consumes time at the phase of population initialization which is being reduced by ODE and other time-consuming phase is candidate solution which is being optimized by the proposed technique.

The concept of opposite numbers used to replace random number which were generated uniformly in some dynamic intervals, have a great contribution to optimization process. Also, opposition concept implies on Crossover for fasten the optimization. So, left over part of proposed algorithm is reserved identical. As per the acceleration rate values, it show 12% and 40% reduction in speed with respect to ODE and DE.

In Fig. 1(a) and Fig. 1(b), acceleration rate of various benchmark functions is plotted. Here AR 1, AR 2 and AR 3 indicating NFE_{DE}/NFE_{CODE}, NFE_{ODE}/NFE_{CODE} and NFE_{DE}/NFE_{ODE} respectively.

Fig. 1(a) gives the values for F_1 - F_{10}. Note that AR 1 dominant the whole graph, after that AR 2 takes place and then AR 3 come into the picture. So it is proven that the proposed algorithm is robust relating to DE and ODE. Also Fig. 1(b) is shown the values for F_{11} - F_{20}. Here the dominant nature of AR 1 is decreased by 30% but still it is working well. On the other hand AR 3 outperforms AR 2 in most of the cases. Subsequently exploration of acceleration rate says that the performance of CODE is much better than the existing algorithm.

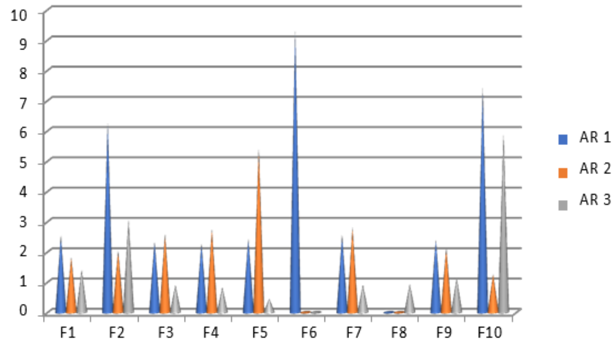

Fig. 1(a) Relation of different Acceleration Rate for F_1 - F_{10}

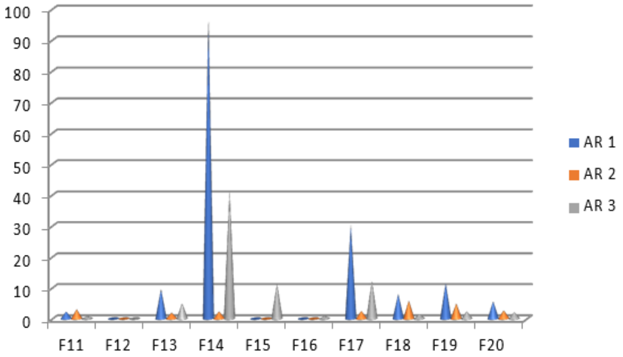

Fig. 1(b) Relation of different Acceleration Rate for F_{11} – F_{20}

For impartial comparison coding has been done again of all existing algorithm with same parameter settings. In valuation, CODE expresses better convergence than DE as well as with ODE. According to [8-9], Classic mutation strategy of DE that is DE/rand/1/bin is used for mutation operation as it is the best mutation strategy for DE, ODE and also for CODE and OBL is utilized in crossover operation for the exploitation purpose of search space.

TABLE II show the comparison of Differential Evolution Algorithm and Opposition Base Differential Evolution Algorithm with proposed algorithm that is Cross-Opposition Based Differential Evolution (CODE) in regards to the mean value of fitness function and standard deviation.

It is difficult to incorporate all function evaluation graphs in limited space. So, some of them are included to understand the flow of convergence. Fig. 2 – Fig. 7 indicate comparative analysis of Differential Evolution, Opposition Based Differential Evolution and Cross-Opposition Based Differential Evolution. The graphs are plotted for the best run of Sphere function, Rosenbrock's valley, Rastrigin function, Griewangk function, Ackley's Path and Hyper-ellipsoid function respectively. The resultant figures show the astounding improvement in NFEs (Number of function evaluation).

The convergence rate of the algorithm has been increase exponentially by incorporating the feature of OBL in crossover.

VII. CONCLUSION

The paper uses the concept of OBL in crossover section for the acceleration purpose of DE. Differential Evolution (DE) has a huge area for exploration as it is simple and robust. Furthermore, the paper proposes a new algorithm named as Cross Opposition Based Differential Evolution(COBDE) algorithm wherein the concept of Opposition based learning(OBL) is hybridized in Differential Evolution. First, OBL is embedded at the phase of population initialization and other is on crossover. The results are analysed with two types of DEs and shows the vigorous performance in relations of convergence, average number of function evaluation and population size.

The summarization of the investigational results is that CODE perform far better that existing algorithm like DE and ODE with respect to the convergence speed. The exploitation capacity of a given technique is flawless. No matter what the population size and dimensions of the problem.

The proposed algorithm has lots of possibilities for optimization. Various steps can be modified and hybridized to optimize the algorithm like different population initialization techniques, strategy of DE, smarter mutation method.

REFERENCES

1. Storn R. and Price K., Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, in Journal of global optimization, 1997, vol. 11, no. 4, 341–359.
2. Rahnamayan S., Tizhoosh H. R. and Salama M. M. A., Opposition based differential evolution algorithms, in IEEE Congress on Evolutionary Computation, 2006, 2010–2017.
3. Tizhoosh H. R., Opposition-based learning: A new scheme for machine intelligence, in Proceedings – International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet, 2005, vol. 1, 695–701.
4. Rahnamayan S., Tizhoosh H. R. and Salama M. M. A., Opposition based differential evolution algorithms, in IEEE Congress on Evolutionary Computation, 2006, 2010–2017.
5. Rahnamayan S., Tizhoosh H. R. and Salama M. M. A., Opposition-based differential evolution for optimization of noisy problems, in: 2006 IEEE World Congress on Computational Intelligence (CEC-2006), Vancouver, BC, Canada, 2006, 6756–6763 .
6. Rahnamayan S., Tizhoosh H. R. and Salama M. M. A., Opposition based differential evolution, IEEE Transactions on Evolutionary Computation, 2008, vol. 12, no. 1, 64–79.
7. J. Brest, S. Greiner, B. Boˇskovic´, M. Mernik, and V. ˇ Zumer, “Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems,” IEEE Trans. Evol. Comput., vol. 10, no. 6, pp. 646–657, Dec. 2006.
8. J. Sun, Q. Zhang, and E. P. K. Tsang, “DE/EDA: A new evolutionary algorithm for global optimization,” Inf. Sci., vol. 169, pp. 249–262G.

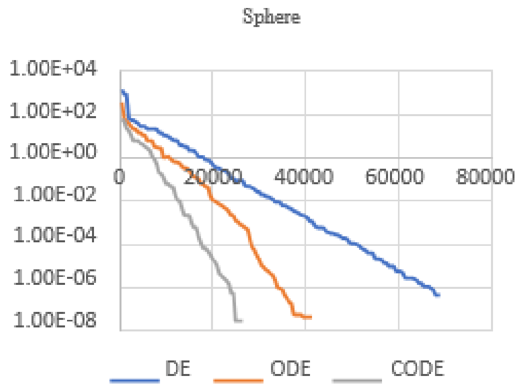

Fig. 2 Comparative Analysis of Sphere function for exploitation

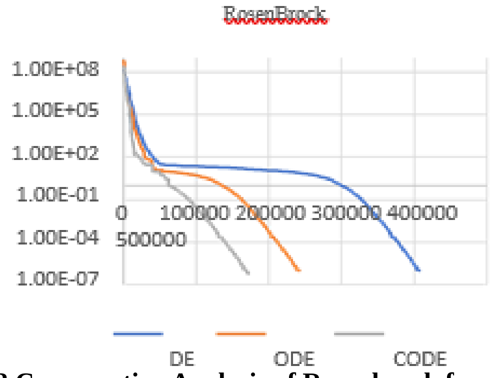

Fig. 3 Comparative Analysis of Rosenbrock function for exploitation

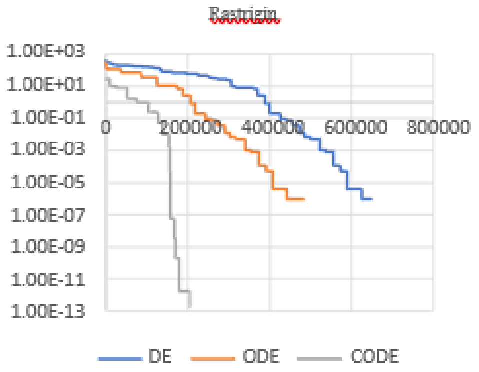

Fig. 4 Comparative Analysis of Rastrigin function for exploitation

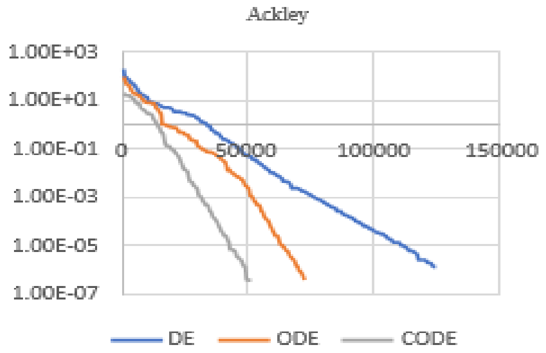

Fig. 5 Comparative Analysis of Ackley function for exploitation

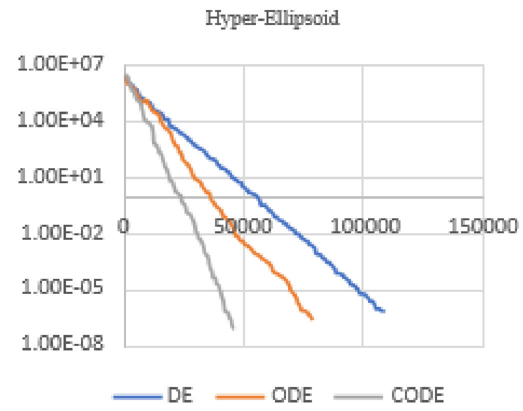

Fig. 6 Comparative Analysis of Hyper-Ellipsoid function for exploitation

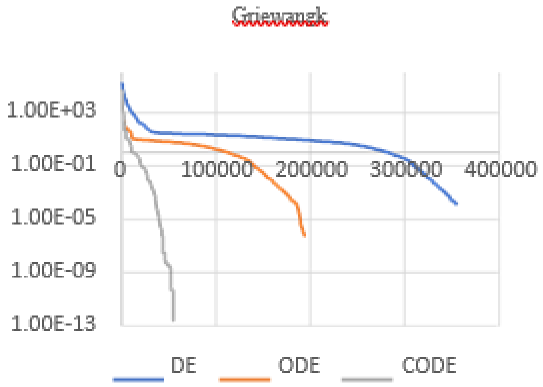

Fig. 7 Comparative Analysis of Griewangk function for exploitation

Retrieval Number B2283078219/19©BEIESP
DOI: 10.35940/ijrte.B2283.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table II Comparison of CODE with DE and ODE in terms of Number of Function Evaluation (NFEs), Mean (M) and Standard Deviation (STD)

Function		DE			ODE			CODE			AR		
		NFE	Mean	STD	NFE	Mean	STD	NFE	Mean	STD	NFEDE/NFECODE	NFEODE/NFECODE	NFEDE/NFEODE
F1	Ackley	127228	1.72E-06	3.07E-07	91410	-0.0695	1.2406	**50162**	2.22E-08	1.68E-07	2.536342251	1.822295762	1.391838967
F2	Alpine	487810	-161.293	2.5836	159252	-0.8869	5.2611	**78096**	0.06879	3.1243	6.246286622	2.039182545	3.063132645
F3	Axisph	77940	5.96E-07	1.37E-07	86442	-0.0254	0.7375	**33296**	3.74E-07	3.00E-05	2.34082172	2.596167708	0.901645034
F4	Brown	65792	6.49E-07	1.35E-07	79536	0.4002	0.5134	**28880**	-1.22E-05	6.90E-05	2.278116343	2.75401662	0.827197747
F5	Chung	66954	1.62E-07	5.63E-08	148656	-0.4622	5.1678	**27518**	5.73E-05	0.0019	2.433098336	5.402136783	0.450395544
F6	Cosmix	255242	-27	3.97E-07	-	-	-	**27484**	5.59E-05	0.0024	9.286930578	-	-
F7	DeJong	41874	1.82E-07	6.82E-08	46132	0.0127	0.1218	**16390**	-2.07E-04	0.003	2.554850519	2.814643075	0.907699644
F8	Dixon	**97310**	0.6667	1.33E-07	104750	0.5304	1.3404	-	-	-	-	-	0.928973747
F9	Expo	54468	-1	1.26E-07	47776	0.0224	0.0454	**22578**	3.05E-05	9.83E-05	2.412436885	2.116042165	1.140070328
F10	Griewangk	79654	6.36E-07	1.17E-07	13592	1.22E-07	6.94E-06	**10723**	8.47E-07	2.06E-06	7.428331624	1.267555721	5.860359035
F11	Hyper	121906	6.44E-07	1.35E-07	162762	15.571	138.0575	**53592**	-3.71E-07	7.87E-06	2.27470518	3.03705777	0.748983178
F12	Quartic	-	-	-	-	-	-	-	-	-	-	-	-
F13	Rastrigin	346572	3.67E-07	5.11E-07	69854	7.12E-07	8.65E-07	**36558**	8.28E-07	1.09E-07	9.480059084	1.910771924	4.961376585
F14	Rosenbrock	4101298	7.47E-07	2.24E-07	99678	3.24E-08	7.12E-07	**42356**	8.90E-07	8.24E-07	96.82920956	2.35333837	41.14546841
F15	Salomon	1857230	0.1019	0.0141	164982	0.5162	4.1039	-	-	-	-	-	11.25716745
F16	Schwefel	**90988**	6.46E-07	2.51E-07	111004	0.36	4.8099	-	-	-	-	-	0.819682174
F17	Sphere	870400	-1.39E-05	7.30E-05	71644	0.0121	0.2854	**28790**	-4.26E-06	8.19E-05	30.23271969	2.488502952	12.14895874
F18	Step	134452	3.05E-05	3.65E-05	98532	-0.0574	4.5051	**16910**	-0.0161	0.4179	7.951034891	5.826848019	1.364551618
F19	Step2	211834	5.13E-06	4.79E-05	91566	-0.501	5.8645	**18504**	0.0376	0.2329	11.44801124	4.94844358	2.31345696
F20	Sum Square	197608	4.67E-06	2.18E-05	96884	-0.1183	1.619	**35702**	4.77E-07	2.62E-05	5.534928015	2.713685508	2.039635027

AUTHORS PROFILE

Shweta Sharma B.Tech in Information Technology, M.Tech in Computer Science Engineering and presently pursuing her Ph.D. in Hybridization of Differential Evolution and Shuffled frog leaping Algorithm and their application from Amity University Rajasthan, India. Her research areas are Evolutionary & nature inspired algorithm and their applications for optimization.

Dr. Ashwani Kumar Yadav has done his Ph.D. from Amity University Rajasthan. He did his M. Tech. in VLSI and Embedded System Design and B. Tech. in Electronics & Communication Engineering from reputed technical institutes of Haryana in India. His research interests include image processing, VLSI design and optimization techniques. He has published various research papers in reputed Journals and refereed international conferences. He is also serving as editorial board member and reviewer of reputed journals and conferences.

Dr. Deepak Sinwar is working as Assistant Professor (Senior Scale) in the Department of Computer and Communication Engineering, School of Computing & Information Technology at Manipal University Jaipur, Jaipur, Rajasthan, India. He has received his Ph.D and M.Tech degrees in Computer Science and Engineering; and B.Tech with honors in Information Technology. His research is focused on Data Mining, Machine Learning and Pattern Recognition. He is having 8 years of experience of teaching various courses of UG/PG level at different Institutes. He has supervised 11 students of M.Tech (CSE) dissertation work and many students at Bachelor's level for their Minor/ Major project work. He has published more than 30 papers in various International Journals, National/ International conferences of repute. He is also working as Reviewer for Concurrent Engineering- Research and Applications (SCI) and Advances in Science, Technology and Engineering Systems (ASTES) Journal (USA)- SCOPUS. He has participated in many Short-Term Courses (STCs)/ Faculty Development Programme (FDPs)/ Workshops/ MOOCs. He is a member of ACM, ISTE (LM), CSTA, IAENG and Internet Society.

.

Bhagyashri Naruka , received the B. E. and M. Tech. degree from the University of Rajasthan and Banasthali University, in 2008 and 2010, respectively. She has more than 5 years of rich experience in teaching and joined Amity University Rajasthan in August 2010. Also, she is pursuing her Ph. D. in the field of Computational intelligence. Her areas of interest are in the field of Differential algorithm, Genetic Algorithm, and Statistical Analysis. She has published a plethora of research papers in national and international Journals & conference proceedings and also attended the number of workshops and conferences.

Dr. Vaishali did her Ph.D. from Amity University Rajasthan. She has completed her initial education from various reputed educational institutes of Haryana in India. She completed her B. Tech (Information Technology) and M. Tech. (Computer Science & Engineering) in 2007 and 2010 respectively. Her areas of interest are Evolutionary algorithm, and their real world applications. She has published a number of research papers in Journals of repute and in refereed international conferences. She is also contributing as reviewer and editorial boards of a few reputed journals