IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 3, September 2023, pp. 1352~1359
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i3.pp1352-1359

Analysis of new differential evolution variants to solve multi-modal problems

Amit Ramesh Khaparde[1], Ramesh Poondi Sundarasamy[2], Sathiyaraj Rajendran[3], Amrita Ticku[4], Arunkumar Palanichamy[5]
[1]Computer Science and Engineering, G. B. Pant DSEU Okhla –I Campus, New Delhi, India
[2]Computer Science and Engineering, Vel Tech Rangarajan Dr. Saguntala R&D Institute of Science andTech, Chennai, India
[3]Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Bengaluru, India
[4]Computer Science and Engineering, Bharati Vidyapeeth College of Engineering, New Delhi, India
[5]Artificial Intelligence and Data Science, Sri Eshwar College of Engineering, Coimbatore, India

Article Info

Article history:

Received Jul 04, 2022
Revised Sep 21, 2022
Accepted Oct 11, 2022

Keywords:

Differential evolution algorithm
Evolutionary computation
Multi modal problems
Uni-modal problems

ABSTRACT

Differential evolution algorithm (DE) requires diversified population to solve multi- modalproblems. DE supports non-distributed population. DE versions include differential evolution algorithm with best selection method and species evolution (DESBS) and differential evolution algorithm with hierarchical fair competition modal (HFCDE). This article analyzed the efficiency of HFCDE and DESBS to solve the multi-modal s' problems. HFCDE and DESBS support non-distributed population structured. HFCDE starts with set of feasible solution then it distributed them in the different hierarchy. HFCDE provides the fair competition. DESBS is another semi distributed, differential evolution algorithm.It starts with set of feasible solutions (population). Later it identifies the niches and create the sub-groups within population. Both HFCDE and DESBS have outperformed the other variant of state-of-art variants of DE. Here, the performance of DESBS and the HFCDE are rigorously tested on the multimodal problems. The success of DESBS over HFCDE in multi-modal difficulties managed to overcome the phenomena of elitism to resolve the complex problems, it has been observed that DESBS performs better than HFCDE in complex multi-modal scenarios.

Corresponding Author:

Amit Ramesh Khaparde
Department of Computer Science and Engineering, G. B. Pant DSEU Okhla – I Campus
OkhlaIndustrial Estate Phase III, New Delhi, India
Email: khaparde.amit@gmail.com

1. INTRODUCTION

Differential evolution (DE) algorithm has successfully solved continuous spaced optimization problems. These problems are non-linear and non- differential in nature. DE is easy to implement and strong algorithm. It has three control parameters. It supports parallel computation. It has high convergence speed. DE has application in every domain of engineering optimization problems. Apart from theses advantages, DE has internal architecture problems that caused DE unproductive, on complex large-scaled optimization problem.

DE has three performance deciding parameters number of population (NP), crossover probability (Cr), and scaling factor (F). And, the parameter setting is a problem. This problem caused other problems like convergence or stagnation. Parameters control the diversity of population during evolution. DE literature has the parameter setting rules and self-adaptive parameter techniques have partially resolved parameter setting issue. But still, a problem independent and self -adaptive (as per problem landscape) parameter adjustment technique is required to improve DE performance. The mutation process distinguishes DE from other class of evolutionary algorithms. It has explorative and exploitative strategies. The mutation techniques decide the convergence speed and parameter values. The imbalanced selection techniques exert low selection pressure and caused premature convergence/stagnation. The performance of mutation operator/strategy is relay on the nature of problem– Example - "DE/rand/1/exp" retains the population diversity and DE/current-to-random/1 not . The self-governing, sensible (explorative in inception and exploitative toward end) mutation strategy can solves the problem. tandard DE provides panmictic central population structure. It has greedy selection process. Hence, it has high convergence speed but it loses the genetic information. The lost genetic material is useful to solved the complex multi-optima problems. The performance of DE may be improved by retaining the genetic material without effecting the convergence speed.

The hierarchical competition modal (HFC) structures the population into the various hierarchical (level). All population members are structured into levels, as per fitness level. Then, every level is parallelly evolved by DE. HFC enables fair competition as per societal system, i,e, the strong individuals never dominate the entire population and every individuals participate in offspring generation. The newly generated offspring are allocated to respective fitness level.

As the HFCDE preserves the genetic material, conserving the non-performing population members during the evolution can decrease the DE performance. To avoid this, HFCDE has replacement method. In natural world, a birth and demise are part of life . In HFC modal , the non performing individuals of lowest level are exchanged by new population individuals by – replacement method. The experimental results of HFCDE (differential evolution algorithm with hierarchical competition modal) shows that replacement method is not mandatory for uni- modal problems but for multi-modal problems [1]. The authors have proposed two variants of HFCDE called HFCDE-R and HFCDE-wR are for uni-modal and multi modal problems, respectively.

Differential evolutionary algorithm with species and best vector selection (DESBS) [2] uses best determination method (BDM) in order to recognized the niches. The species are formed around the niches. All species are evolved by standard differential evolutionary algorithm. Timely, the performance of all species are evaluated. The non- performing species are merged into the adjacent performing species. The species formation, merging and BDM exploit the search space.

DESBS consider the population as the collection of individuals. Here individuals are represented as vectors. DESBS consider male and female individual based on fitness values of vectors. The fittest individuals (above the threshold) are considered as a female individua, whereas, remaining are considered as a male individual. The best identification method (BDM) determines the fittest individuals (Female) in the initial population. DESBS allows all the individuals to produce the offspring. The standard differential evolutionary algorithm is used for evolution.

DESBS consider the organisms/individuals of common characteristic features species. And every organism has own role in nature, DESBS consider this as a niche. DESBS collects all the individuals having similar characteristics into a niche (groups). The performance of each niche is evaluated at specific time interval and non-performing species are merged to nearby species.

Considering the DESBS and HFCDE both consider the initial population then structured the population into species and hierarchy, respectively. Both DESBS and HFCDE evaluate the performance of population structure. The non-performing species in DESBS and hierarchy in HFCDE are merged to nearby species or hierarchy. The DESBS does not guarantee fair competition whereas HFCDE guarantee the fair competition. DESBS does not retains the genetic material whereas HFCDE retains. In this article we have performed the comparative analysis of these two population variants of DE on the uni-modal and multi-modal functions.

This manuscript is divided into the five sections. Differential evolution algorithm (DE), Differential evolution algorithm with hierarchical fair competition modal (HFCDE) and differential evolution algorithm with best selection method and species evolution (DESBS) are briefly explained in second section. DE has always attracted the community of researchers and researchers have worked to improve the DE performance, in third section, the recent related work on the DE performance improvement is elaborated. The performance of HFCDE and DESBS is rigorously tested and the results are explained in section four. Finally, the work is concluded the last section.

2. METHODS

In this section, the DE algorithm, DE algorithm with HFCDE, and DE algorithm with the DESBS are discussed inthe detail. Both provide the facility of distributed structured population. Both algorithms explore the search space very rigorously. One of the major differences of HFCDE and DESBS that HFCDE supports the retention of genetic material, whereas, DSBES does not retains genetic material. HFCDE does not allows the best individuals to dominate the population, whereas, DESBS allows best individuals to dominate the population. As a result, it is necessary to evaluate the performance of both to determine whether or not the preservation of genetic material is worthwhile.

For evolution, DESBS and HFCDE used the standard DE algorithm. DE is a simple, fast, parallel evolutionary algorithm and used to deal with complex non-combinatorial problems. DE is a four-stage cycle named generation, mutation, crossover, and selection. Its mutation operation distinguishes it from other evolutionary algorithms.

2.1. Differential evolution algorithm

DE begins with the generation NP vectors, in (1) is used to generate the predefined 'NP' number of randomly selected vectors. Here, $X_{i_{max}}, X_{i_{min}}$ are the upper and the lower bounds of decision variables. Each member (vector) represented by $X_{i,j} = X_{1i,j}, X_{2i,j}, \ldots X_{Di,j}$, whereas i=1,2 … NP.‘D’ is number of variables. And, ‘j’ shows the respective generation.

$$X_{i,j} = X_{i_{min}} + \text{rand}\,[0,1]\left(X_{i_{max}} - X_{i_{min}}\right) \tag{1}$$

$$M_{i,j} = X_{r1,i} + F\left(X_{r2,j} - X_{r3,j}\right) \tag{2}$$

In mutation a new vector called mutant vector is creates by using (2).r_1, r_2, r_3 belongs to 1. . . NP and mutually different numbers. F > 0 within range [0, 2] known as scaling factor. Recombination (crossover) caused diversity in population and represented by (3). Muation in standard differential evolution algorithm makes it different from another evolutionary algorithm.

$$V_{i,j} = \begin{cases} M_{i,j}\,, \text{if}\left(\text{r\_and}(i) \leq \text{Cr or } i = \text{r\_and}_i(j)\right) \\ X_{i,j}\,, \text{otherwise} \end{cases} \tag{3}$$

In (3), the r_and(i) is a random number generator, it generates number between (0,1]. ‘Cr’ is crossover probability between (0,1]. randi(j) is index between (1,D]. $M_{i,j}$ is the mutant vector generated in mutation phase by (2). In (4) represents greedy selection, here, the fittest vector between trail vector and target vector are retained for the next generation. $V_{i,j}$ is the donor vector generated in crossover phase. $X_{i,j}$ is present generation vector, whereas, $X_{i+1,j}$ is the next generation vector,

$$X_{i,j+1} = \begin{cases} V_{i,j}\,, \text{if}(f(V_{i,j}) \leq f(X_{i,j})) \\ X_{i,j}\,, \text{otherwise} \end{cases} \tag{4}$$

2.2. Hierarchical fair competition modal with hierarchical fair competition modal

The HFCDE generates the initial population as per (1). HFCDE uses population division method (PDM) to divide the initial population into the various levels. PDM compute the admission threshold (AT) before deciding the levels. The individuals are allocated to respective levels based on AT. AT for lowest level, I,e, f_{adm}^{0} is highest fitness value of the entire population. The AT for the other pre-defined levels are computed by the (5).

$$f_{ads}(L) = fu + (L-1) * \frac{std + fu - fmin}{level - 2} \tag{5}$$

Whereas, Fu is average fitness of the population. std is the standard division of population and fmin is the minimal fitness value of the function. Level represents number of level. HFCDE uses DE for evolution. The new members generated during evolution are migrated to corresponding levels using migration method. There are various migration methods are available in DE literature. The recognition of migration method based on type of problems can be an area of research? Finally, HFCDE replaces all non-performing employees with new employees using the Replacement Method. The replacement method replaces a predetermined number of non-performing individuals with new randomly generated individuals.

2.3. Distributed differential evolution algorithm with best selection method and species evolution algorithm

DESBS starts with generation of initial population using (1). Like PDM of HFCDE, DESBS has BDM. BDM recognized the predefined number of the fittest individuals from the initial population. These individuals are called as best members. The best members are act liked niches and the species (sub-population groups) are formed around these niches. BDM uses Euclidian distance to decide the members in the species.

Each species are evolved using standard differential evolution algorithm parallelly. At last, the non- performing species are merged to nearby species after specific interval. The process is continued till stopping criteria does not satisfied

3. RELATED WORK

The balanced exploration and exploitation has enhanced the DE performance. The proposed mutation operator called Hemostatic operator [3], increases the diversity by utilizing the best members in inception. It caused no stagnation at end and provides promising results. The variant of DE/current-to-best called DE/current-to-gr-best [4] has improved the DE. An adaptive differential evolution with unsymmetrical mutation [5] provide balance exploration and exploitation

Differential evolution algorithm never uses probability density function (PDF). But, the Gaussian-parent-centric blend crossover (PBX-alpha) or differential evolution algorithm with a new improved DE algorithm [6], advance crowding-based differential evolution (CRDE) [7] used the probability density functions to explore the search space. The population structure of the multi-scaled DE [8] is sub-population. In the crossover operator, it employs a covariance learning enabled coordinate system. Factors make it more efficient than other alternatives. These two It has solved the global optimization problems efficiently and successfully.

Proximate mutation operator [9] computes Euclidian distance and ranked based mutation operator [10] computes fitness value in mutation operator. This computation improved DE performance. DE has many mutation operators, but, operationally DE uses them as static, i.e, DE cannot change the mutation strategy during evolution. The intended results were obtained through runtime mutation strategy selection using historical information or problem landscape. The dynamic trail vector selection by composite differntial evolution algorithm (CoDE) [11] has given the desired results. Sub- population structured with covariance matrix based novel variamt of differntial evolution algorithm [12] has speedup the performance and given the desired results.

Ensemble strategies in compact differential evolution called EPSDE [13] generates the combinations of different control parameters and mutation strategies. EPSDE creates the child vector using created parameter combinations then conducts tournament to produce the offspring for the next generation. EPSDE is more successful than compared to variants to solve complex problems.

The new distributed differential evolution [14] uses a time-efficientmodal called hierarchical island mode. Author proposed master-slave system for two-level parallel computation. The co-operative hierarchical structured system provides the exhaustive search of solution space.

Modified multi-objective and self-adaptive DE, (MMOSADE) [15], has successfully solved multi-objective design issuesof nuclear power systems. Nuclear plants are using MMOSADE. The new Memetic framework with Alopex local search [16] with DE (MFDEALS) the control global exploration of solution space. DE's performance has improved as a result of the proposed framework. It performs better than the other metaheuristic algorithms. The new framework was put through its paces using both standard and adaptive DE.

The Dichotomy-based parameter adaptation DE (DPADE) [17] is a novel self-adaptive parameter scheme. It shrinks the parameter spaces by utilizing information for parameter setting. DPADE has efficiently solved complex problems.

A multi-population inflationary differential evolution algorithm with adaptive local restart [18] is a novel self-adaptive evolutionary algorithm. It makes use of a multi-population structure and furthermore, DE is combined with local search. To avoid local minimal, local restart procedures and global restart procedures are employed.

SaDSE [19], is a new self-adaptive dual-strategy with differential evolution algorithm. It performs well on large-dimension problems. For exploitation and exploration, SaDSE employs "DE/best/2" and "DE/rand/2," respectively. It offers balanced exploration and exploitation. In exploration, it employs a self-adaptive scaling factor.

The hybrid of the moth-flame optimization and extreme learning machine (MFO-ELM) algorithm [20] enhances the performance of traditional extreme learning machine (ELM) by a hundred percentage (100%). MFO-ELM is a classification algorithm and it 80 % more efficient than the compared algorithms.

Fuzzy logic is a rule-based system to address optimization problems. The parameterization of fuzzy arithmetic is a challenging task. Phishing detection using fuzzy arithmetic is an attractive area of research. A comparative analysis of metaheuristics for fuzzy arithmetic parameter setting is performed on two novel datasets and performance is evaluated on four parameters [21]. One of the applications of metaheuristic algorithms are clustering of web documents, anEmpirical analysis of clustering techniques using GloVe and density – based algorithm is conducted and it has been found that the density-based spatial clustering of applications with noise (DBSCAN) and density peaks clustering (DPC) are the most promising density-based clustering techniques [22].

Genetic algorithm (GA) is used to improve many soft computing techniques. The GA has improved the k-nearest neighbour feature extraction from computed tomography (CT) scan images [23]. Formation control of a multi-agent system is a constraint satisfaction problem and it has been unccessfully resolved by particle swarm optimization [24].

4. RESULTS AND DISCUSSION

Both the algorithms, DE algorithm with HFCDE and DE algorithm with best vector selection and species evolution, begin with random predefined number of viable solutions - initial population. Both HFCDE and DESBS divide the population in levels and subgroups, respectively. The standard differential evolution (SDE) algorithm is employed in both HCDE and DESBS for evolution for generating new members. The newly generated individuals are further distributed at their respective hierarchies and groups in HCDE and DESBS, respectively. At the end, the performance of levels and subgroups are evaluated and non-performing levels and groups are merged into nearby levels or subgroups. The DESBS and HFCDE are run for twenty (20) times on all twenty-five (25) problems out of which five (5) problems are uni-modal and twenty (20) are multi-modal. All problems are minimization problems, hence the minimal value i.e. ‘min’ and standard division (spread of final population) is considerd as parameters for the comparison.

4.1. Experimental results

The parameter setting for both HFCDE and DESBS are the same as suggested by authors and represented in Table 1, represents that the given parameter is not applicable for a respective algorithm. The authors have proposed two variants of HFCDE called HFCDE-wR and HFCDE-R for both uni-modal and multi-modalfunctions, respectively. Hence, HFCDE-wR is compared with DESBS for uni-modal problems and HFCDE-R with DESBS for multi-modal problems.

To check the null hypothesis, the two-trail Wilcoxon rank-sum test is used and the significance level is set at 0.05. “There is no difference exists between HFCDE (HFCDE-wR for unimodal functions and HFCDE-R for multimodal functions) and DESBS”, in accordance with the null hypothesis. There are three cases, one is if the null hypothesis is rejected and the HFCDE (HFCDE-wR for uni-modal functions and HFCDE-R for multi-modal functions) outperforms the DESBS in a statistically significant way, the cases are characterized as positive ("+"). Secondly,if it is rejected and the DESBS significantly outperformed the HFCDE (HFCDE-wR for unimodal functions and HFCDE-R for multimodal functions), the cases are characterized as negative ("-"); and thirdly, if the null hypothesis is accepted but also there is no discernible performance difference, the cases are characterized with “=”. The CEC 2005 [25] test suite is used for experimentation.

The experimental outcomes of HFCDE-wR vs. DESBS are presented in Table 2 and HFCDE-R vs DESBS are presented in Table 3. Two parameters-the mean (minimum value) and the STD (standard deviation after population evaluation) are used to evaluate the results. Figure 1 demonstrates the results' executive summary.

Table 1. Parameter’s setting

S. No.	Parameters used	HFCDE	DESBS
		Values	
1	Scaling factor (F)	0.5	0.5
2	Number of variables (D)	30	30
3	Initial population (NP)	300	300
4	Crossover rate (Cr)	0.9	0.9
5	Max function evaluation	3000000	3000000
6	No. of levels	3	*
7	Replacement probability (Rp)	0,0.5	*
6	No. of individuals to replace	50 %	*
7	No. of species to be formed (R)	*	4
8	Merging interval (M)	*	15

Table 2. HFC-DE/wR Vs DESBS

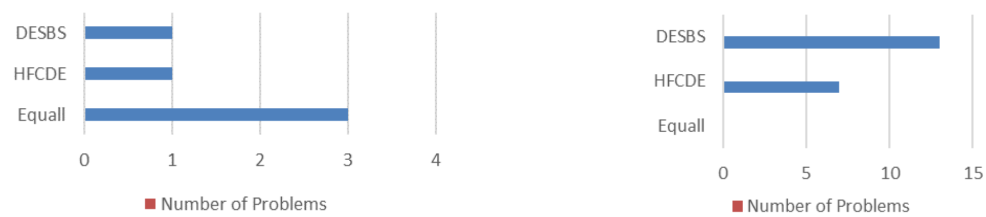
Problem number	HFC-DE/wR		DESBS		Sign
	Min	STD	Min	STD	
1	0	0	0	0	=
2	0	0	0	0	=
3	29455.86	50915.054	50915.054	53812.754	+
4	0	0	0	0	=
5	58.56625	26.007016	26.007016	23.987682	-

Table 3. HFC-DE-R Vs DESBS

Problem number	HFCDE / Rp		DESBS		Sign
	Min	STD	Min	STD	
1	0.000997	0.000908	0	0	-
2	0.000931	0.000823	1.130142	0.516809	+
3	20.85372	0.036228	20.845565	0.044809	+
4	74.51388	20.89869	0	0	-
5	154.194	38.87278	26.666991	4.210192	-
6	36.36954	1.736027	29.09254	1.768606	-
7	444298.4	185500.2	18548.362	2938.4635	-
8	14.20333	0.969519	1.363809	0.096251	-
9	13.2871	0.160706	12.9299	0.151972	-
10	631.3784	117.3084	202.91252	2.008797	-
11	181.9051	46.49874	99.452783	106.13368	-
12	210.9485	39.5949	125.62366	98.787843	-
13	815.8842	0.002144	816.15367	0.156606	+
14	815.8843	0.00173	816.1648	0.200726	+
15	815.8842	0.002031	816.1395	0.152328	+
16	856.2283	0.01193	857.12859	0.606353	+
17	499.9001	0.000046	500.05744	0.64505	+
18	863.6347	0.352885	864.2592	0.870808	-
19	208.5016	0.053614	210.11687	0.684087	-
20	208.4965	0.036887	215.95424	40.987819	-

4.2. Discussion

There are total five uni-modal functions in test suite. The experimental findings indicate that both algorithms perform equally well in problems 1, 2, and 4, whereas, in problem 3 HFCDE (HFCDE-wR) is performed better than DESBS and in problem 5 DESBS has outperformed HFCDE (HFCDE-wR). The complexity of test problem in considered test suite is increases from 1 to 5 in uni-modal problem and from 1 to 20 in multi-modal problem. There is total 20 (twenty) multi – modal problems in test suite. HFCDE (HFCDE-R) are performed better in problem number 2, 3, 13, 14, 15, 16, 17 than DESSBS, whereas, in rest, I,e, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 18, 19, 20 DESBS has outperformed HFCDE (HFCDE-R). It is clear from the results that in uni-modal complex problems DESBS has performed better than HFCDE) HFCDE-wR) and DESBS has outperformed HFCDE (HFCDE-R) in thirteen problems out of twenty problems. The graphical representation of dominance of DESBS over HFCDE is represented in the Figure 1.



DESBS VS HFCDE in Unimodal Problems DESBS VS HFCDE in Multi- modal Problems

Figure 1. Summary of experimentation

5. CONCLUSION

The DE with HFCDE and DE with are two alternative population-structured differential evolution algorithms that are compared on both uni-modal and multi-modal functions. Both algorithms are population-structured semi-distributed algorithms. Standard differential evolution algorithm is employed by both methods. While DESBS condemns elitism, HFCDE does. HFCDE is more computationally expensive than DESBS because it uses three additional variables over the conventional differential evolution algorithm (SDE), whereas DESBS uses two additional variables. According to experimental findings, DESBS is more effective than HFCE at solving complicated multi-modal problems.

REFERENCES

[1] A. R. Khaparde *et al.*, "Differential evolution algorithm with hierarchical fair competition model," *Intelligent Automation and Soft Computing*, vol. 33, no. 2, pp. 1045–1062, 2022, doi: 10.32604/iasc.2022.023270.

[2] A. R. Khaparde, M. M. Raghuwanshi, and L. G. Malik, "A new distributed differential evolution algorithm," *International Conference on Computing, Communication and Automation, ICCCA 2015*, pp. 558–562, 2015, doi: 10.1109/CCAA.2015.7148456.

[3] S. Prabha and R. Yadav, "Differential evolution with biological-based mutation operator," *Engineering Science and Technology, an International Journal*, vol. 23, no. 2, pp. 253–263, 2020, doi: 10.1016/j.jestch.2019.05.012.
[4] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012, doi: 10.1109/TSMCB.2011.2167966.
[5] E. C. Shi, F. H. F. Leung, and J. C. Y. Lai, "An adaptive differential evolution with unsymmetrical mutation," *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 1879–1886, 2011, doi: 10.1109/CEC.2011.5949844.
[6] R. Zhou, J. Hao, H. Cao, and H. Fan, "An empirical study on differential evolution algorithm and its several variants," *Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011*, vol. 6, pp. 3266–3271, 2011, doi: 10.1109/EMEIT.2011.6023708.
[7] L. Chen and L. Ding, "An improved crowding-based differential evolution for multimodal optimization," *2011 International Conference on Electrical and Control Engineering, ICECE 2011 - Proceedings*, pp. 1973–1977, 2011, doi: 10.1109/ICECENG.2011.6057739.
[8] Y. Du, Y. Fan, X. Liu, Y. Luo, J. Tang, and P. Liu, "Multiscale cooperative differential evolution algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, 2019, doi: 10.1155/2019/5259129.
[9] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011, doi: 10.1109/TEVC.2010.2083670.
[10] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066–2081, 2013, doi: 10.1109/TCYB.2013.2239988.
[11] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011, doi: 10.1109/TEVC.2010.2087271.
[12] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 89–99, 2013, doi: 10.1109/TII.2012.2198658.
[13] R. Mallipeddi, G. Iacca, P. N. Suganthan, F. Neri, and E. Mininno, "Ensemble strategies in compact differential evolution," *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 1972–1977, 2011, doi: 10.1109/CEC.2011.5949857.
[14] M. L. Tardivo, P. Caymes-Scutari, G. Bianchini, and M. Méndez-Garabetti, "Hierarchical parallel model for improving performance on differential evolution," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, 2017, doi: 10.1002/cpe.4087.
[15] Y. Yang *et al.*, "A modified multiobjective self-adaptive differential evolution algorithm and its application on optimization design of the nuclear power system," *Science and Technology of Nuclear Installations*, vol. 2019, 2019, doi: 10.1155/2019/1041486.
[16] M. Leon, N. Xiong, D. Molina, and F. Herrera, "A novel memetic framework for enhancing differential evolution algorithms via combination with alopex local search," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 795–808, 2019, doi: 10.2991/ijcis.d.190711.001.
[17] L. Cui, G. Li, Z. Zhu, Z. Ming, Z. Wen, and N. Lu, "Differential evolution algorithm with dichotomy-based parameter space compression," *Soft Computing*, vol. 23, no. 11, pp. 3643–3660, 2019, doi: 10.1007/s00500-018-3015-2.
[18] M. Di Carlo, M. Vasile, and E. Minisci, "Multi-population inflationary differential evolution algorithm with adaptive local restart," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, vol. 24, no. 5, pp. 632–639, doi: 10.1109/CEC.2015.7256950.
[19] M. Duan, H. Yang, S. Wang, and Y. Liu, "Self-adaptive dual-strategy differential evolution algorithm," *PLoS ONE*, vol. 14, no. 10, 2019, doi: 10.1371/journal.pone.0222706.
[20] O. A. Alade, R. Sallehuddin, and N. H. M. Radzi, "Enhancing extreme learning machines classification with moth-flame optimization technique," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 1027–1035, 2022, doi: 10.11591/ijeecs.v26.i2.pp1027-1035.
[21] N. S. Nordin *et al.*, "A comparative analysis of metaheuristic algorithms in fuzzy modelling for phishing attack detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1146–1158, 2021, doi: 10.11591/ijeecs.v23.i2.pp1146-1158.
[22] S. M. Mohammed, K. Jacksi, and S. R. M. Zeebaree, "A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 1, pp. 552–562, 2021, doi: 10.11591/ijeecs.v22.i1.pp552-562.
[23] R. A. Nugroho, A. S. Nugraha, A. Al Rasyid, and F. W. Rahayu, "Improvement on KNN using genetic algorithm and combined feature extraction to identify COVID-19 sufferers based on CT scan image," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, pp. 1581–1587, 2021, doi: 10.12928/TELKOMNIKA.v19i5.18535.
[24] A. M. Hasan and S. M. Raafat, "Optimized formation control of multi-agent system using PSO algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, p. 1591, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1591-1600.
[25] P. N. Suganthan *et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report 2005005, IIT Kanpur, India*, 2005.

BIOGRAPHIES OF AUTHORS

Dr. Amit R. Khaparde received the Bachelors of Engineering, M. Tech and Ph. D from renowned universities. Presently, he is working as Assistant Professor in Dept. Of Computer Science and Engg in G. B. Pant DSEU Okhla – I campus. He is working in area of soft computing and machine translation from last twelve years. He has many research publications in various international, national journals and conferences. He can be contacted at email: khaparde.amit@gmail.com.

Dr. Ramesh Poondi Sundarasamy has completed BE, M.E.,M.B.A.,MCSI., MISTE. And Ph.D. from renowned universities. He has more than 13 years of teaching experience and his research interests include image processing, theoretical computer science, distributed system and cloud computing. He has published 10 papers in reputed international SCI/ Scopus indexed journal and presented 21 papers in reputed IEEE/Springer/ACM international conference and won several best paper awards. He can be contacted at email: psramesh7@gmail.com.

Dr. Sathiyaraj Rajendran has completed his Ph.D. at Anna University Chennai. His research interests lie in the area of Big Data Analytics, AI, and IoT. He has collaborated actively with researchers in several other disciplines of computer science, particularly traffic prediction systems and intelligent systems. Has more than 9 years of experience. He has authored more than 25 publications and filed 5 patents. He can be contacted at email: rsr026@gmail.com.

Ms. Amrita Ticku perusing her PhD degree in Computer Science and Engineering from MVN University Palwal, Haryana. She received her M. Tech. Computer Science and Engineering from MDU University Haryana, India, in 2011.She has more than 10 years of Research and Teaching Experience in the domain of Computer Science and Engineering. Currently she is an Assistant Professor in Computer Science and Engineering, Bharati Vidyapeeth college of Engineering Delhi, India. Her research interest includes internet of things (IoT), block chain and AI & ML. she has published papers in several international conferences and journals. She can be contacted at email: amritaticku27@gmail.com.

Mr. P. Arunkumar has completed his UG and PG from renowned universities. He has sixteen year of experience and working area is computer network. He has many international and national journal and conference publications. He can be contacted at email: arunkumar.p@sece.ac.in.