

Peregrine Preying Pattern based Differential Evolution

Sanjay Jain¹, Vivek Kumar Sharma¹, and Sandeep Kumar³

¹ Amity University Rajasthan, Jaipur- 302039 India,
jainsanjay17@yahoo.co.in

² CHRIST (Deemed to be University), Bangalore-560029, India
sandpoonia@gmail.com

Abstract. The differential evaluation (DE) algorithm is an evolutionary algorithm. It is a popular metaheuristics that efficiently solved various complex optimization problems. This paper proposed modification in DE, motivated by the flying pattern of peregrine falcon while preying and named as peregrine preying pattern based DE algorithm (P3DE). In P3DE, the flying pattern of peregrine improves the exploration potential, while preserving the novel exploitation potential of the DE. The competence, precision, robustness and trustworthiness of the projected P3DE algorithm is analyzed while simulating it over 12 composite benchmark functions of diverse characteristics. The results of P3DE compared with DE and Artificial Bee Colony algorithm to prove its competitiveness for selected problems.

Keywords: Evolutionary Algorithm, Logarithmic Spiral, Metaheuristics, Nature Inspired Algorithm, Optimization

1 Introduction

Storn and Price [16] developed a simple and fast algorithm in year 1995 namely Differential Evolution. The fundamental concept in DE is to make use of the vector differences for perturbing vector solutions. DE is a stochastic meta-heuristic. DE fits into the class of Evolutionary Algorithms (EAs). A number of characteristics like trial vector development procedure (discussed in section 2) make use of the information about direction and distance from present population to engender a fresh trial vector is significantly contrasting with other existing evolutionary strategies. In case of DE the mutation is applied first and then crossover applied while in all other EAs, most of the time a trial vector generated using crossover in first step and then one offspring produced using the mutation operation.

DE is very popular among researchers specially working in the field of optimization as it has a number of advantages over other population-based algorithms. But similar to other stochastic algorithms DE also has some drawback and researchers are frequently working to enhance the efficiency and reliability of this evolutionary algorithm. A few recent and most popular variants of DE

are discussed in [2] with appropriate applications. DE performs better than the many other competitive algorithms like Genetic Algorithm (GA) [5, 9] and the Particle Swarm Optimization (PSO) [8] for considered numerical benchmarks [28] and many more new algorithms [17]. DE has proved its superiority over other algorithm in various applications of science, management, engineering and technology like chemical engineering [13], mechanical engineering design [18], machine intelligence, pattern recognition [15], robot path planning [22] and signal processing [3]. Now a days DE algorithm is one of the most accepted strategies in area of engineering optimization, machine intelligence and cybernetics.

The evolution of population in DE is driven by variation and selection. The deviation process is absolutely accountable for the exploration of different areas of the probable search region and exploitation of the finest solution is done by the process of selection. A number of new study shows that sometimes DE is not able to achieve global optimum [12]. In this paper, to set up an appropriate trade-off between diversification and convergence of the population in DE, a natural phenomenon of specific flying pattern exhibited by peregrine falcon while preying is implemented with DE and named as peregrine preying pattern based DE algorithm (P3DE).

2 Overview of DE Algorithm

The general notation for DE is: $DE/x/y/z$, here DE stands for Differential Evolution, x represents the strategy for selection of target vector, number of differential vector used perturbation of x denoted by y , and z represent the crossover technique employed. Based of selection of x , y and z [16] various variant of DE are available. The most popular DE scheme $DE/rand/1/bin$ and also used in this paper. It select the target vector randomly and use binomial crossover with one differential vector. The DE recognised among researchers as it is very simple, robust, easy in implementation and has wide class of applicability to real world problems [21].

2.1 Mutation

The mutation operator generate a trial solution by doing some random changes in trial solution. The working of mutation operator to engender a trial solution u_i from the parent solution x_i is defined as follow:

- Select a target vector $x_{i_1}(g)$ in such a way that i and i_1 are not same.
- Identify two random solutions, x_{i_2} and x_{i_3} in such a way that i , i_1 , i_2 and i_3 are different.
- The trial vector computed by mutating target vector using equation 1

$$u_i(g) = x_{i_1}(g) + F \times (x_{i_2}(g) - x_{i_3}(g)) \quad (1)$$

where mutation scale factor $F \in [0, 1]$ control the escalation of the differential variation [4].

2.2 Crossover

DE apply uniform crossover to engender offspring $x'_i(g)$ using parent vector, $x_i(g)$ and the trial vector, $u_i(g)$ as follows:

$$x'_{ij}(g) = \begin{cases} u_{ij}(g), & \text{if } j \in J \\ x_{ij}(g), & \text{otherwise.} \end{cases} \quad (2)$$

where a set of crossover points denoted by J , j^{th} element of $x_i(g)$ vector denoted by $x_{ij}(g)$. It is supposed that if $j \in J$, the trial parameter is selected from the mutant $u_{ij}(g)$.

2.3 Selection

The selection operator decides between trial vector and target vector. It decides the individual to compute the trial vector for the mutation and precisely choose the best for the next generation, between the trial vector and their predecessors based on their fitness value. If fitness of trial vector is lower than target vector then it takes over from the target vector for next generation. Contrarily, the target vector $x_i(g)$ remains there for next generation i.e. best fitted individual selected for next generation.

$$x_i(g+1) = \begin{cases} x'_i(g), & \text{if } f(x'_i(g)) < f(x_i(g)). \\ x_i(g), & \text{else.} \end{cases}$$

In order to mend the results of basic DE, researchers have proposed a number of variants of DE. It has been observed by Storn and Price in [16] that the best suitable range for value of F is $[0.5, 1]$ and $[5D, 10D]$ is the most appropriate range of the value of NP , where, considered problem is of D dimension.

Most of the researchers focused on identification of best suitable value for control parameters (F and CR) but very few researchers tried to identify the best suitable size of the population (NP) for performance enhancement. New variants of DE proposed by Teo [26] and Sharma et al. [19] based on the concept of self adaption in populations and dynamic scaling. They suggested self adaption in order to avoid manual parameter setting. Some new strategies also incorporated in DE like: opposition based strategy [11], fitness based strategy [20], convex mutation [23], memetic search [10] and position update [6]. Some modifications in population selection suggested by researchers [24].

3 Peregrine Preying Pattern based DE Algorithm

The peregrine falcon is one of the most intelligent raptor from the falconidae family of birds that hunts and feeds on rodents and other animals. Peregrine falcons has two important skills, first is flight with speed over 200 mph and visual sharpness that are very helpful while attacking their prey. Peregrine falcons are

found worldwide. The peregrine follows a particular path while chasing prey and it is similar to logarithmic spiral. The studies reveal that peregrine falcons follow a logarithmic spiral path to attack their prey [14, 1]. The peregrine flees along a logarithmic spiral route to avoid the conflict between vision and aerodynamics [27].

This paper presents an improved mutation strategy inspired by the path followed by peregrine falcon during attack on prey i.e. logarithmic spiral. It is a self-similar spiral curve which formed by peregrine falcon while preying [1]. As the solution search process of DE algorithm is highly depends on a combination of randomly selected vectors and a difference vector (refer equation 1). Hence, there are more chances to omit the true solution in case of high value of scaling factor F and the difference vector. Therefore, here a new approach to decide scaling factor proposed, which helps the current best solution in the swarm to exploit the potential solutions in its vicinity.

The equation of logarithmic spiral is shown in equation 3. It is the locus of points going apart from a centre point with a invariable speed besides a line which rotates with fixed angular velocity proportional to the positions with time of a point. Unvaryingly, in polar coordinates (r, θ) as illustrated by the equation 3 [1].

$$r = a \times e^{b\theta} \quad (3)$$

Here, a and b are real numbers while e denotes the base of natural logarithms. The parameter a control the turning of the spiral and the distance between consecutive turnings managed by b as shown in Figure 1.

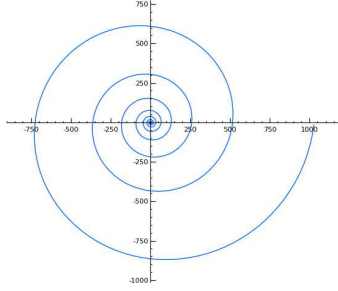


Fig. 1: Logarithmic spiral [1]

In P3DE, the best solution in the current pool of solution is permitted to modernize their position adaptively. The new mutation approach is developed by taking inspiration from the logarithmic spiral as shown in equation 4.

$$u_i(g) = x_{i_1}(g) + F_{P3DE} \times (x_{i_2}(g) - x_{i_3}(g)) \quad (4)$$

$$\text{where, } F_{P3DE} = 2 \times \text{sign} \times U(0, 1) \times \left(1 - \frac{\text{Iter}}{TI}\right) \times e^{\frac{\sin(\text{Iter})}{\text{Iter}}} \quad (5)$$

Here, $Iter$ is the counter for search iteration, TI represents total number of iterations, $sign$ is the addition or subtraction sign which depends on the fitness of the new solution. The scale factor is calculated using equation 5 which is developed by modification of logarithmic spiral equation. In this equation $a = 2 \times sign \times U(0, 1) \times (1 - \frac{Iter}{TI})$ and $\theta = \sin(Iter)$. The step size is used to provide distance to the best individual during the search process.

4 Experimental results and discussion

The performance of newly developed variant (P3DE) evaluated over thirteen unbiased standard problems and analysed here in terms of precision, efficacy and trustworthiness. A set of 12 mathematical optimization problems with diverse degree of complexity are selected to confirm the proficiency of the planned P3DE algorithm. All the selected functions are continuous in nature. These problems are minimization problems and solutions of the most of the functions does not exists on the origin, diagonal and axis.

4.1 Experimental setting

In order to check the performance of P3DE, initial population (N) was taken 50 and number of iterations (itr) taken as 4000. The newly developed P3DE algorithm is compared with basic DE [25] and ABC [7] for the purpose of assessment to evaluate the efficiency, robustness and reliability. Simulation results of the newly developed P3DE algorithm and the considered algorithms are presented in terms of success rate (SR), average number of function evaluations (AFE) and mean error (ME) as mentioned in Tables 2, 1 and 3 respectively. Standard deviation (SD) also measured for analysis. The considered algorithms are examined with the following experimental setting:

4.2 Results Analysis of Experiments

To evaluate the efficiency, an AFE based comparison is depicted out in Table 1. It can be easily observed from this table that the AFE of P3DE is less for most of the functions i.e. the newly developed P3DE is converge to optima faster than the other considered algorithms. Further, the Table 2 shows the results of successful runs over 100 simulations. A simulation is considered successful, if the algorithm achieves optima at the level of acceptable error. The Table 2 elucidate that the newly developed P3DE is more reliable for most of the benchmark functions in terms of success rate as compared to the considered algorithms. The robustness and accuracy of the newly developed P3DE algorithm is measured by the ME as depicted in Table 3. This table exhibits competitiveness of the purposed algorithm in terms of accuracy for the considered Test Problems (TP).

Table 1: Comparison based on AFE.

TP	ABC	DE	P3DE
Sphere Function	23218	23226	22002
Griewank Function	76412	64616	34147.5
Michalewicz	100011.18	173182.5	83717.5
Cosine Mixture	100041.88	30999.5	22303
Step function	17442.73	33200	10459
Inverted cosine wave	100010.73	178795.5	44205.5
Colville Function	99696.08	30623.5	24117
Kowalik Function	91466.63	66029	28883.5
2D Tripod function	5594.96	19454.5	6394
Shifted Griewank	100008.44	153104.5	37651.5
Meyer and Roth	25858.6	16355	4298
Sinusoidal Problem	100037.57	200050	127026.5

Table 2: Comparison of success rate.

TP	ABC	DE	P3DE
Sphere Function	100	100	100
Griewank Function	68	81	99
Michalewicz	0	19	64
Cosine Mixture	0	96	100
Step function	100	91	100
Inverted cosine wave	0	15	99
Colville Function	1	87	95
Kowalik Function	16	69	96
2D Tripod function	100	92	99
Shifted Griewank	0	30	100
Meyer and Roth	98	93	100
Sinusoidal Problem	0	0	65

Table 3: Comparison of mean error.

TP	ABC	DE	P3DE
Sphere Function	7.86E-06	8.77E-06	9.06E-06
Griewank Function	4.36E-03	2.37E-03	8.30E-05
Michalewicz	6.36E+00	4.02E-02	1.46E-02
Cosine Mixture	2.62E+00	7.40E-03	8.99E-06
Step function	0.00E+00	3.20E-01	0.00E+00
Inverted cosine wave	6.62E+00	8.43E-01	5.25E-03
Colville Function	2.26E-01	6.82E-02	1.53E-02
Kowalik Function	1.90E-04	6.35E-04	1.11E-04
2D Tripod function	6.34E-05	8.01E-02	1.01E-02
Shifted Griewank	9.07E+01	1.29E-02	7.95E-06
Meyer and Roth	1.95E-03	1.96E-03	1.95E-03
Sinusoidal Problem	2.96E+00	3.58E+00	9.83E-02

Table 4: Comparison of AR.

TP	ABC	DE
Sphere Function	1.05526	1.05563
Griewank Function	2.23770	1.89226
Michalewicz	1.19462	2.06865
Cosine Mixture	4.48557	1.38992
Step function	1.66772	3.17429
Inverted cosine wave	2.26240	4.04464
Colville Function	4.13385	1.26978
Kowalik Function	3.16674	2.28604
2D Tripod function	0.87503	3.04261
Shifted Griewank	2.65616	4.06635
Meyer and Roth	6.01642	3.80525
Sinusoidal Problem	0.78753	1.57486

4.3 Statistical Analysis

The comparison of P3DE with DE and ABC is done on the basis of AFE, SR, and ME. The results in Tables 1, 2 and 3 show that P3DE is very effective for all considered test problems while these problems are of different nature. After observing these results it may be concluded that P3DE is able to balance the process of exploitation and exploration very effectively. The boxplot [29] analysis of AFE for P3DE, ABC and DE have been presented in Figure 2 to denote the distribution of outcomes. It is clearly visible through the boxplots analyses of the results as shown in Figure 2 that P3DE outperforms the considered algorithms. While observing the boxplots of success rate, the median of P3DE is high whereas interquartile range is low as compared to the other considered algorithms which proves the reliability of the P3DE over the compared algorithms.

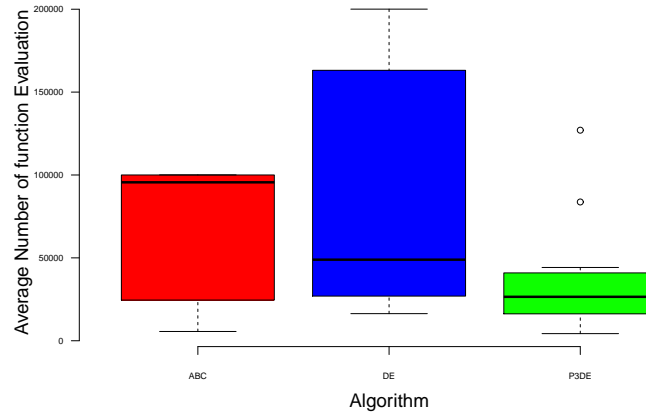


Fig. 2: Boxplots graph for AFE

Further, a fair comparison in terms of the convergence speed is done through AR analyses on the AFEs of the considered algorithms. The AR is calculated by equation 6.

$$AR = \frac{AFE_{ALGO}}{AFE_{P3DE}}, \quad (6)$$

where, $ALGO \in \{DE \text{ and } ABC\}$. It is clear from equation 6 that the AR will be high for the algorithm having fewer AFEs and vice-versa. The calculated AR is

presented in Table 4. While observing the Table 4, the value of AR is more than one for most of the function which shows that the newly developed P3DE is fast convergent algorithm as compared to the other considered algorithms.

5 Conclusion

A new modification in DE algorithm is suggested in this paper and named as Peregrine Preying Pattern based DE (P3DE) algorithm. The newly anticipated strategy is inspired by a unique pattern followed by peregrine falcon while chasing its prey that is depicted as logarithmic spiral. The exploration capability of the DE algorithm is improved by the logarithmic spiral search process. To assess the anticipated P3DE algorithm, 12 benchmark functions are selected for experiments. Results proved that P3DE will be a better choice for optimization problems.

Bibliography

- [1] <https://en.wikipedia.org/wiki/logarithmicspiral> accessed on 02/07/2018, July 2018.
- [2] U.K. Chakraborty. *Advances in differential evolution*. Springer Verlag, 2008.
- [3] S. Das and A. Konar. Two-dimensional iir filter design with modern search heuristics: A comparative study. *International Journal of Computational Intelligence and Applications*, 6(3):329–355, 2006.
- [4] A.P. Engelbrecht. *Computational intelligence: an introduction*. Wiley, 2007.
- [5] J.H. Holland. *Adaptation in natural and artificial systems*. Number 53. University of Michigan press, 1975.
- [6] Sanjay Jain, Sandeep Kumar, Vivek Kumar Sharma, and Harish Sharma. Improved differential evolution algorithm. In *Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS), 2017 International Conference on*, pages 627–632. IEEE, 2017.
- [7] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.
- [8] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
- [9] Sandeep Kumar, Sanjay Jain, and Harish Sharma. Genetic algorithms. In *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, pages 27–52. Chapman and Hall/CRC, 2018.
- [10] Sandeep Kumar, Vivek Kumar Sharma, and Rajani Kumari. Memetic search in differential evolution algorithm. *arXiv preprint arXiv:1408.0101*, 2014.
- [11] Sandeep Kumar, Vivek Kumar Sharma, Rajani Kumari, Vishnu Prakash Sharma, and Harish Sharma. Opposition based levy flight search in differential evolution algorithm. In *Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on*, pages 361–367. IEEE, 2014.
- [12] J. Lampinen and I. Zelinka. On stagnation of the differential evolution algorithm. In *Proceedings of MENDEL*, pages 76–83, 2000.
- [13] P.K. Liu and F.S. Wang. Inverse problems of biological systems using multi-objective optimization. *Journal of the Chinese Institute of Chemical Engineers*, 39(5):399–406, 2008.
- [14] John W Lorimer. Curved paths in raptor flight: Deterministic models. *Journal of theoretical biology*, 242(4):880–889, 2006.
- [15] MGH. Omran, AP Engelbrecht, and A Salman. Differential evolution methods for unsupervised image classification. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 966–973. IEEE, 2005.
- [16] K.V. Price. Differential evolution: a fast and simple numerical optimizer. In *Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American*, pages 524–527. IEEE, 1996.

- [17] J Rajpurohit, TK Sharma, A Abraham, and Vaishali. Glossary of meta-heuristic algorithms. *Int J Comput Inf Syst Ind Manag Appl*, 9:181–205, 2017.
- [18] T. Rogalsky, S. Kocabiyik, and RW Derksen. Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal*, 46(4):183–190, 2000.
- [19] Harish Sharma, Jagdish Chand Bansal, and KV Arya. Dynamic scaling factor based differential evolution algorithm. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*, pages 73–85. Springer, 2012.
- [20] Harish Sharma, Jagdish Chand Bansal, and KV Arya. Fitness based differential evolution. *Memetic Computing*, 4(4):303–316, 2012.
- [21] Prashant Sharma, Harish Sharma, Sandeep Kumar, and Jagdish Chand Bansal. A review on scale factor strategies in differential evolution algorithm. In *Soft Computing for Problem Solving*, pages 925–943. Springer, 2019.
- [22] Prashant Sharma, Harish Sharma, Sandeep Kumar, and Kavita Sharma. Black-hole gbest differential evolution algorithm for solving robot path planning problem. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 1009–1022. Springer, 2019.
- [23] TK Sharma et al. Asynchronous differential evolution with convex mutation. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, pages 915–928. Springer, 2016.
- [24] VP Sharma, HR Choudhary, S Kumar, and V Choudhary. A modified de: Population or generation based levy flight differential evolution (pglfde). In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on*, pages 704–710. IEEE, 2015.
- [25] R. Storn and K. Price. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Global Optimiz*, 11:341–359, 1997.
- [26] J. Teo. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 10(8):673–686, 2006.
- [27] VA Tucker, AE Tucker, KATHY Akers, and JH Enderson. Curved flight paths and sideways vision in peregrine falcons (*falco peregrinus*). *Journal of Experimental Biology*, 203(24):3755–3763, 2000.
- [28] J. Vesterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1980–1987. IEEE, 2004.
- [29] D.F. Williamson, R.A. Parker, and J.S. Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916, 1989.