

# Parameter optimization of Cuckoo Search Algorithm for Multi Dimensional Function Optimization Problem

Mohammad Shafiul Alam

Ahsanullah University of

Science and Technology

Samiha Sara Prima

Ahsanullah University of Science and Technology Dhaka-1208, Bangladesh Sanonda Datta Gupta

Ahsanullah University of Science and Technology Dhaka-1208, Bangladesh Jannatun Razia

Ahsanullah University of Science and Technology Dhaka-1208, Bangladesh

# Dhaka-1208, Bangladesh

ABSTRACT The Cuckoo Search Algorithm is a recently developed nature inspired meta heuristic algorithm, which is established on the breeding behavior of Cuckoo species. Cuckoo search can be applied on a large variety of optimization problems. The main advantage of this search algorithm is its simplicity and better performance than many other agent or population based meta heuristic algorithms. The algorithm uses only one controlling parameter p, which makes it easier to implement and control. This parameter p, combined with the random walk mutations implemented by Lévy Flights, can control the performance and degree of exploration and exploitation of the algorithm. In this paper we have conducted a few experiments on Cuckoo Search algorithm with Lévy flights to discover the necessary conditions needed for the better performance of the algorithm. For this purpose we have taken different values of the controlling parameter p and observed the performance of the algorithm on benchmark problems, as well as its exploration and exploitation characteristics over different groups of benchmark functions.

# **Keywords**

Cuckoo search algorithm, function optimization, Lévy flights mutation, parameter optimization.

# 1. INTRODUCTION

The aim of any optimization algorithm is to converge to a solution which is the best candidate following some criteria. Optimization problems may include minimizing or maximizing a function by having a set of input values and finding out for which value the convergence criteria is satisfied. In real world problems we apply optimization programming in every walk of our life. First we describe and formulate the optimization problem correctly then the task is to find the best (sometimes just near optimum) solution that meet certain conditions. Among many existing optimization algorithms (e.g., [1]-[11]), we have focused on one population based meta heuristic algorithm - the Cuckoo Search algorithm [1]. Meta heuristic algorithms are stochastic optimization procedures that allow random perturbations of existing candidate solutions to produce optimal or near optimal solutions to complex problems where the best (i.e., global optimum) solution can very hardly be found. Meta heuristic algorithms are widely used and recently many algorithms of this kind have been developed (e.g., [1]-[11]) by getting inspired from natural and biological phenomena.

The Cuckoo search meta heuristic algorithm performs better than efficient genetic algorithms and many other swarm intelligence algorithms. The algorithm is inspired by the obligate brood parasitism of cuckoo species that lays eggs in the nest of some other host species. It was developed by Xin She Yang and Suash Dev in the year 2009. Later Yang and Deb enhanced the performance of the algorithm by using Lévy flights for random search style. Lévy flights were introduced because flight behavior of some birds and fruit flies have similar characteristics with it. In this paper we have used cuckoo search algorithm with Lévy flights. Cuckoo Search is well known for its simplicity having only one control parameter *p*, which is the discovery rate of cuckoo egg by the host bird. The objective is to find the perfect parameter value that gives best performance and optimum degree of exploration and exploitation. The performance test is conducted on low and high dimensional unimodal and multimodal functions for each parameter value.

The rest of the paper organized as follows. Section 2 describes ideas of cuckoo search algorithm with Lévy flights and brief pseudo code. In section 3 experimental studies and results are presented. Finally section 4 draws conclusion and presents future research directions based on this work.

# 2. CUCKOO SEARCH ALGORITHM

The cuckoo is a brood parasite, they lay eggs in the nest of other birds. Group of female cuckoos favor a particular host species nest and laving eggs that match of that species in color and pattern. When a cuckoo it finds a host nest with incubating eggs, which makes parasitizing impossible, it destroys the whole nest. This usually forces the host birds to lay fresh eggs giving the cuckoo a chance to parasite the nest. If a host bird discovers the eggs are not their own they will either get rid of those alien eggs or simply abandon its nest and build a new nest elsewhere. Parasitic cuckoos always choose a nest where the host birds just laid its own eggs. In general cuckoo eggs hatch slightly earlier than their host eggs. Once the cuckoo chick is hatched the first instinct action it will take is to evict the host eggs by blindly propelling the eggs out of the nest increasing cuckoo chick's share of food provided by its own host bird. Studies also show that a cuckoo chick can also mimic the call of host chicks to gain access to more feeding opportunities. In this paper the version of cuckoo search algorithm used follows the following three rules.

- One cuckoo will lay one egg on one host nest at a time. The host nest is chosen at random.
- Only fittest nests with high quality of eggs will survive to the nest generation.
- The number of available host nests is fixed and discovery rate of each cuckoo egg by the host bird is *p* ∈ [0, 1]. In this case the host bird will abandon the



#### nest upon discovery and build a new one.

To generate each new solution Lévy flight is implemented in the Cuckoo search algorithm. Lévy flights are a random walk whose step length is produced from Lévy probability distribution. The term Lévy flight was first introduced by Benoit Mandelbrat who used Cauchy flight where step size distribution follows Cauchy distribution and Rayleigh flight where step size is a normal distribution. Lévy flight is implemented in chaos theory and stochastic search algorithms. Lévy flights works in two steps - firstly, the direction of random walk is chosen and secondly, steps are originated at that direction which follows Lévy distribution. This random walk has a heavy tailed random walk. In nature many birds and animals follow food hunting paths those can be represented using Lévy flights. This flight gives efficient results in natural flight data. Sometimes Lévy flight creates new offspring far from its parent, thus ensuring the search algorithm does not get stuck around the locally optimal points. The Lévy flight is implemented in CSA as follows.

$$x_i^{t+1} = x_i^t + \alpha * \text{Lévy}(\lambda)$$
(1)

#### Procedure Cuckoo Search Algorithm (n)

Determine fitness of each candidate solution of the population and define maximum iteration number

#### While (I < MaxIterarion)

Choose a nest, *i* at random and find its solution using Lévy flights by Mantegna's algorithm and evaluate its fitness,  $F_i$  using objective function.

Choose another nest, j randomly and calculate its fitness  $F_j$  using objective function.

If  $(F_i > F_j)$  then

replace nest *j* by nest *i*'s new solution

#### End if

Cuckoo eggs are discovered by host bird with the probability, *p* and these nests are abandoned and new nests are made.

The best solutions survive and becomes initial population in the next generation.

#### **End While**

#### Figure 1: Pseudo code of the Cuckoo Search Algorithm

Where  $\alpha > 0$  is the step size of the random Lévy walk. The rate and degree of explorations and exploitations rate of the Cuckoo search algorithm depends on the control parameter, *p*. The Pseudo code is presented in Fig. 1, where the parameter *n* represents the size of initial population of host nests.

### **3. EXPERIMENTAL STUDIES**

This experiment uses 26 benchmark test functions to evaluate the performance of Cuckoo Search Algorithm by taking different values of control parameter p. Lower value of the parameter means less discovery rate of cuckoo eggs by the host birds and higher value means more probability of discovering the cuckoo egg. In Table 1 the benchmark functions we used are included along with dimensions, search space, characteristics and function value at the global maximum. Eight of them are unimodal (function no.1–8) and the remaining 18 (function no. 9-26) are multimodal. The low dimensional functions are the function nos. 2, 5, 6, 11, 12, 15, 19, 21-26, while the rest are high dimensional. Functions of bowl shape (i.e.,  $f_{16}$ ,  $f_{18}$ ), plate shape (i.e.,  $f_4$ ), valley shape (i.e.,  $f_{17}$ ,  $f_8$ ), steep, ridges (i.e.,  $f_2$ ,  $f_{19}$ ) and many local minima (i.e.,  $f_{18}$ ) are also present in the benchmark suite. The Easom Function is unimodal and its global minimum is confined to a small area compared to the entire search space whereas Dixon price function is unimodal high dimensional and gives maximum among all unimodal function as seen from Table 2. For multimodal functions, the search algorithms should have both exploitative and explorative characteristics so it can explore local minima without being stuck and find the global optimum. Hundreds of local minima can occur in multimodal functions even if it is low dimensional. For example, the Cross-In function has multiple global minima with many local minima. Drop wave function is a highly complex multimodal function to evaluate. Griewank function has many widespread local minima regularly distributed and evaluated on hypercube. Michalewicz function has many local minima which equals to the factorial of dimension of the function and its steepness and ridges make it difficult to search. Since the increase of dimensions exponentially increase the number of local minima, the high dimensional (i.e.,  $D = 20 \sim 30$ ) multimodal functions are usually extremely difficult to be optimized by the search algorithms. Here the search algorithm may get stuck in local optima bypassing the global optimum and give erroneous results. In case of low dimensional functions, the locally minimal points are regularly separated by wide regions, so they are also difficult to explore without being trapped. Here low dimensional functions are function nos. 2, 5, 6, 9, 11, 12, 15, 17, 19-24.

The results from the experimental studies are given in Table 2 where initial population size is 20 and the number of maximum generation is 500 for each benchmark function. The minimum error values are also included in Table 2 which is the difference between actual function value at global minima and the value the algorithm found at the end of its execution. The mean absolute error value is calculated in Table 3 for both unimodal and multi modal functions. The key observations from the experiment are stated in the following points.

- Out of five value that we took for control parameter p, the p = 0.1 gives best result on 17 functions, 0.3 gives best result on 16 functions becoming the second best scorer. This means the lower value of the discovery rate p of cuckoo eggs by the host bird usually produces better performance. The p = 0.1 means only 10% of nest containing cuckoo eggs were discovered by host bird. So the number of abandoned nest is least (only 10%).
- On unimodal and multi modal functions with low dimensions, the search algorithm performs excellent. For example, the function nos. 2, 13, 19, 21–26 gives zero error for most values of the parameter *p*.
- For function nos. 8, 10, 17, 18 all strategies fail and they get trapped around the local optima and fail to converge to global optimum causing premature convergence. For example, Rosenbrock Function (i.e.  $f_{17}$ ) has global minimum inside a long, narrow, parabolic shaped flat valley which is difficult to find. Here, the algorithm gets stuck in local optima for all five parameter values.



- Considering the mean absolute error (MAE) as a performance measurement metric, we found that p = 0.5 gives the least MAE in eight unimodal functions. Among 18 multimodal functions, p = 0.3 gives best result. The second best result is given by p = 0.7. Without considering mean absolute error, we see that p = 0.1 gives the best results in five functions out of the eight and for multimodal function p = 0.3 gives best result on 14 functions.
- On functions with few local minima the low number of dimensions or simpler structure, the parameter value p = 0.1 gives best result. This indicates that the algorithm gets more explorative and can perform fine graining and locate global minimum. As we increase the parameter value, it becomes more and exploitative. The value of p = 0.3 provides good result on 14 test functions and it gives better performance on Rosenbrock and Ackley function than p = 0.1.

Fable 1: Standard Benchmark Functions where D: Dimensionality, S: Search Space, C: Function Cha	RACTERISTICS
WITH THE FOLLOWING VALUES — U: UNIMODAL, M: MULTIMODAL, $f_{min}$ : Function Value at Global Mi	NIMUM.

No	Function	С	D	S	$f_{min}$
$f_1$	Powell	U	24	$[-4, 5]^D$	0
$f_2$	Easom	U	2	$[-100, 100]^D$	-1
$f_3$	Discuss	U	30	$[-100, 100]^D$	0
$f_4$	Sum Series	U	30	$[-10, 10]^D$	0
$f_5$	Zakhazov	U	2	$[-5,10]^{D}$	0
$f_6$	Matyas	U	2	$[-10, 10]^D$	0
$f_7$	High Conditioned Eclipse	U	20	$[-100, 100]^D$	0
$f_8$	Dixon Price	U	30	$[-10, 10]^{D}$	0
$f_9$	Perm	М	4	$[-4, 4]^{D}$	0
$f_{10}$	Rastrigin	М	30	[-5.12, 5.12] <sup>D</sup>	0
$f_{11}$	Beale	М	2	$[-4.5, 4.5]^{D}$	0
$f_{12}$	Booth	М	2	[-10, 10] <sup>D</sup>	0
$f_{13}$	Shekel	М	10	$[0, 10]^{D}$	-10.53
$f_{14}$	Griewank	М	30	[-600, 600] <sup>D</sup>	0
$f_{15}$	Hartman	М	3	$[0, 1]^D$	-3.862
$f_{16}$	Lévy	М	30	[-10, 10] <sup>D</sup>	0
$f_{17}$	Rosenbrock	М	30	[-5, 10] <sup>D</sup>	0
$f_{18}$	Ackley	М	30	[-32.76, 32.76] <sup>D</sup>	0
$f_{19}$	Michalewicz	М	2	$[0,\pi]^D$	-1.81
$f_{20}$	Rotated Hyper- Ellipsiod Function	М	10	[-65.53, 65.53] <sup>D</sup>	0
$f_{21}$	Six Hump Camel	М	2	[-5, 5] <sup>D</sup>	-1.30
$f_{22}$	Eggholder	М	2	[-512, 512] <sup>D</sup>	-959.64
$f_{23}$	Mccormick	М	2	[-4.5, 8] <sup>D</sup>	-1.91
$f_{24}$	Cross-In-Tray	М	2	[-10, 10] <sup>D</sup>	-2.06



<i>f</i> <sub>25</sub>	Drop-Wave	М	2	$[-5.12, 5.12]^{D}$	-1
$f_{26}$	Shubert	М	2	[-5.12, 5.12] <sup>D</sup>	-186.73

 TABLE 2: Performance of the Cuckoo Search Algorithm on benchmark functions with different values of the control parameter *p*, which determines the rate of discovery of alien eggs. The algorithm was run 500 different times independently on each benchmark function. The best result on each function is marked with boldface font.

Function	Minimum Error Value					Best Performance
No	<i>p</i> = 0.1	<i>p</i> = 0.3	<i>p</i> = 0.5	<i>p</i> = 0.7	<i>p</i> = 0.9	By
$f_1$	0.696	0.106	0.732	1.985	5.108	<i>p</i> = 0.3
$f_2$	0.0	0.0	0.0	0.0	0.06	p = 0.1/0.3/0.5/0.7
$f_3$	4.53e-16	2.04e-14	3.21e-13	7.91e-10	0.098	p = 0.1
$f_4$	1.26	0.84	0.71	2.21	3.44	<i>p</i> = 0.5
$f_5$	2.18e-28	1.52e-25	6.62e–20	1.74e–15	1.97e-07	<i>p</i> = 0.1
$f_6$	1.41e-34	4.86e-24	3.60e-18	2.46e-12	4.76e-06	<i>p</i> = 0.1
$f_7$	2.78e-20	9.27e-17	8.47e-15	1.49e-08	0.0029	<i>p</i> = 0.1
$f_8$	29.68	5.39	2.50	5.80	8.18	<i>p</i> = 0.5
$f_9$	0.027	0.003	0.05	0.01	0.07	<i>p</i> = 0.3
$f_{10}$	84.42	80.93	86.02	89.60	54.85	<i>p</i> = 0.9
$f_{11}$	1.79e-25	4.78e-16	1.61e-14	1.45e-07	0.002	<i>p</i> = 0.1
$f_{12}$	5.44e-20	4.35e-18	6.69e-10	1.68e-05	0.004	<i>p</i> = 0.1
$f_{13}$	0.0	0.0	0.0	0.0	0.03	Except $p = 0.9$ all similar
$f_{14}$	1.43	1.05	1.08	1.10	1.40	<i>p</i> = 0.3
$f_{15}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{16}$	1.28	8.66	13.33	7.60	4.59	<i>p</i> = 0.1
$f_{17}$	599.29	174.74	195.96	189.73	250.22	<i>p</i> = 0.3
$f_{18}$	3.90	3.20	3.58	5.55	6.49	<i>p</i> = 0.3
$f_{19}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{20}$	3.24e-05	5.55e-07	4.27e-05	8.42e-05	0.048	<i>p</i> = 0.3
$f_{21}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{22}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{23}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{24}$	0.0	0.0	0.0	0.0	0.0	All similar
$f_{25}$	0.0	0.0	0.0	0.0	0.001	All similar except $p = 0.9$
$f_{26}$	0.0	0.0	0.0	0.0	0.0	All similar



Value of Control	Mean Absolute Error Value			
parameter, p	Unimodal Function	Multimodal Function		
0.1	3.95	38.35		
0.3	0.79	14.90		
0.5	0.49	16.66		
0.7	1.24	16.31		
0.9	2.10	17.65		

 TABLE 3: Performance comparison of Cuckoo Search Algorithm with different values of the control parameter, p. The best performances are marked with boldface font.

## 4. CONCLUSION

This paper conducts an experiment on the Cuckoo search algorithm to find out which value of the control parameter pcan produce overall best result on complex benchmark functions. Results reveal that if we take value of p from the interval of [0.1-0.3], the algorithm gives excellent overall results. Recent experiments show that the Cuckoo Search algorithm is far better than particle swarm optimization and many existing genetic algorithm variants. Besides, the Cuckoo search is very easy to implement and have only one control parameter *p*. We propose a few suggestions for future research directions. Firstly, the basic Cuckoo search algorithm can be easily hybridized with other genetic and swarm intelligence algorithms, such as the Flower pollination and Simulated annealing Algorithms to improve its performance better. Secondly, the basic Cuckoo search and hybrid Cuckoo search algorithm variants may can employed on many existing discrete, combinatorial and real word optimization problems, including design, scientific and engineering optimization problems, NP hard combinational problems, Data fusion in wireless sensor networks, training neural networks, manufacturing scheduling, Nano electronic technology based operation amplifier and so on.

#### 5. REFERENCES

- [1] X. Yang and S. Deb, "Cuckoo search: recent advances and applications", *Neural Computing and Applications*, vol. 24, no. 1, pp. 169-174, 2013.
- [2] Y. Xu, P. Fan and L. Yuan, "A Simple and Efficient Artificial Bee Colony Algorithm", *Mathematical Problems in Engineering*, vol. 2013, pp. 1-9, 2013.
- [3] X. Yang and X. He, "Bat algorithm: literature review and applications", *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, p. 141, 2013.
- [4] X. Yang, "Flower Pollination Algorithm for Global Optimization", Unconventional Computation and Natural Computation, pp. 240-249, 2012.

- [5] X. Yang, M. Karamanoglu and X. He, "Multi-objective Flower Algorithm for Optimization", *Procedia Computer Science*, vol. 18, pp. 861-868, 2013.
- [6] M. Abdel-Baset and I. Hezam, "A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems", *International Journal of Computer Applications*, vol. 140, no. 12, pp. 10-23, 2016.
- [7] O. Raouf, M. Baset and I. henawy, "A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems", 2014.
- [8] O. Abdel Raouf, I. El henawy and M. Abdel Baset, "A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles", *International Journal of Modern Education and Computer Science*, vol. 6, no. 3, pp. 38-44, 2014.
- [9] B. Nozohour-leilabady and B. Fazelabdolabadi, "On the application of artificial bee colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (PSO) methodology", *Petroleum*, vol. 2, no. 1, pp. 79-89, 2016.
- [10] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen and A. Auger, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Opt.mization", Technical Report, Nanyang Technological University (NTU), Singapore, May 2005 And KanGAL Report #2005005, IIT Kanpur, India.
- [11] B. Liu, Q. Chen and Q. Zhang, J. J. Liang, P. N. Suganthan, "Problem Definitions and Evaluation Criteria for Computationally Expensive Single Objective Numerical Optimization", Technical Report, Zhengzhou University, China and Technical Report, NTU, Singapore, December 2013.