International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 24 (2019) pp. 4434-4440

Cuckoo Search Algorithm based Dynamic Parameter Adjustment Mechanism for Solving Global Optimization Problems

[1,*]**Waleed Alomoush**
[1]*Computer Department, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia*

Abstract

Cuckoo search algorithm CSA is a recent optimization algorithm of swarm intelligence, which has demonstrated powerful outcomes on many optimization issues. nevertheless, it has some limitations such as stuck in local optima and premature convergence especially When solving complicated problems of optimization. Also, the CSA parameters are static during generations time which lead to stuck in local optima and couldn't find the best solutions. In this paper we proposed an improved standard cuckoo search algorithm based dynamic parameter adjustment mechanism called (CSDPA). The CSDPA presents two equation to update the parameters values of steps size and discovery probability during search process. The experiments are tested on ten conventional benchmark functions. Outcomes demonstrate the new CSDPA approach is outperform of the CSA and another CSA variants.

Keywords: Optimization, Swarm Intelligence, Cuckoo Search Algorithm, Lévy Flights

1. INTRODUCTION

Optimization plays a major role in addressing various engineering issues. The purpose of the optimization method is to evaluate the minimum or maximum problem value to be solved, usually known as the fitness function. [1, 2].

Recently, many algorithms are proposed based nature inspired. an example of these algorithms are swarm intelligence, that are a special class of nature inspired algorithms, that have been evolved by implement behaviors of nature [3]. an examples of these algorithms are ant colony algorithm (ACO) [4], particle swarm algorithm (PSO) [5], bat algorithm [6], firefly algorithm (FA) [7] and cuckoo search algorithm (CSA) [8]. The most of Swarm intelligence have been successfully used in many fields such as image processing [9-12], data clustering [13, 14].

One of the most common optimization algorithm is the Cuckoo search algorithm (CSA), that has been widely adapted to a wide range of optimization issues [15, 16]. However, a random selection is used to select a solution that contrasts later with the Lévy flight solution. Therefore, this can be easily stuck in local optima by using random selection when solving complex multimode issues. Even, its used fixed value of parameters. In the initialization step, these values are set and cannot be modified in new generations. The main drawback of this approach is to determine an best solution in the number of iterations [17]. This paper introduce an improved cuckoo search (CSA) based dynamic parameter adjustment mechanism called (CSDPA). CSDPA have dynamic parameteres, which are change during search process, that is lead to aviod trapped into local optima.

The rest of this research paper is design as follows. The related works in section 2 which is dived to orginal CS and varaints of CS, Section 3 introduces the CSDPA approach. Section 4 describes the experimental results. Lastly, in Section 6, conclusion.

2. RELATED WORKS

2.1 behaviour of Cuckoo

Cuckoo birds are a group with a special breeding method that is more hostile than other species of birds. Several kind of cuckoo birds including Ani and Guira lay eggs in the nests of the family. Nevertheless, other eggs can be excluded to increase opportunity to hatch their own eggs. Certain kind take the process of brood parasitism to lay their eggs in other birds ' nests or host nests [8, 18].

In competitive nests where eggs were just laid, the parasitic cuckoos are successful and their accurate period of laying eggs. Its lay single egg that usually hatches faster than the other eggs in the host nest. If this happens, when the eggs have been push outside of the nest, the foreign cuckoo would take out the unhatched eggs of the nest. Such action is intended to reduce the probability of legitime hatching eggs. In addition, by imitating the host birds call, the foreign chick of cuckoo may earn reached to extra eat. If the cuckoo of host finds that any of the eggs is foreign. In this situation the cuckoo either remove of the foreign egg or completely leaves the nest and goes somewhere else to creat other nest [8, 18].

2.2 Lévy flights

Lévy flights are random walks which paths are random and from the distribution of Lévy their path lengths are derived. Such Lévy flights are carried out by animals and insects and are characterized by a sequence of straight flights preceded by abrupt turns of 90 degrees. Lévy flights are much more effective in exploration large-scale search regions compared to normal random walks. This is mostly because the differences in Lévy flights rise much quicker than the normal random walk. [14].

2.3 CSA algorithm

Every egg in a nest is a solution, and a cuckoo egg is a new solution. The objective is to use new and potentially better solutions to change the worst solutions. In the straightforward case, every nest has single egg . The technique can be generalized to more complex cases where each nest has many eggs that represent a group of solutions (Yang 2009; Yang 2010). The CSA is established on three steps and principles:

- Every cuckoo lays an egg at a time and drops it in a nest selected at random;
- The best egg nests (i.e, solutions) of high quality will be dispatched to the next generations.;
- The amount of potential host nests is constant, and a host is likely to find an alien egg Pa ∈ [0, 1] . In this scenario, the host bird may either abandon the egg or leave the nest to creat a new nest in other location (Yang 2009).

For clarity, the last supposition can be estimated by replacing a fraction Pa of the n nests by new nests, with new random solutions. The value or quality of a solution can be related to the bjective function for a problem of maximization. Other aspects of fitness in genetic algorithms can be described in a similar manner to the objective function (Yang 2009). according to the above steps, the important steps of the CS can be illustrated as in the pseudo code, in Figure1.

The following Lévy flight is conducted while generating new solutions xi (t+1) for the ith cuckoo as in Equation 1.

$$x_i(t+1) = x_i(t) + \alpha \oplus levy(\lambda) \quad (1)$$

where α > 0 is the step size to be correlated with the scale of the interest problem. The product ⊕ implies multiplications of the entry-wise. The following equation 2 can be presented a global explorative random walk using Levy flights.

$$le'vy \approx u = t^{-\lambda}, 1 < \lambda \leq 3 \quad (2)$$

where λ is the mean or estimated parameter of the variance of the occurrence over the unit interval. The steps here are basically a random walk system with a strong tail power law step-length distribution. Levy's walk around the optimal solution achieved so far should generate some of the new solutions, which will accelerate local search. Nonetheless, a significant fraction of the alternative solutions must be obtained using far-reaching field randomization and which positions must be far sufficient from the current best solutions.

```
Objective function f (X), X= (f (x1, x2... xd) T
Generate initial population of n host nests
Xi (i=1, 2... n)
While t < Max_itertions do
        Get a cuckoo randomly by Levy flights
        Evaluate its quality/ fitness Fi
        Choose a nest among n (say, j) randomly
        If Fi > Fj then
                replace j by the new solution;
        End If
        Fractions (Pa) of worse nests are
        abandoned and new
        Ones are built;
        Keep the best solutions
        Rank the solutions and find the current best
End While
Postprocess results and visualization
```

Figure1: Pseudo code of the Cuckoo Search Algorithm

2.4 varints of cuckoo search

According to Valian, et al. [17], The CSA parameters are maintained static. This could lead in decreasing the algorithm's effectiveness. An appropriate approach for adjusting the parameters of the CSA is provided to address this problem. Using many well-known benchmark problems, statistical experiments show that in contrast with the solutions achieved by the cuckoo search, the suggested algorithm will find much better solutions.

Mareli and Twala [18] The purposed of this paper is to introduce three novel Cuckoo search (CSA) depending on unfixed parameters for switching. The triple novel CSA algorithms were tested on 10 numerical benchmark functions and their performance compared to CSA algorithms with static and dynamically reducing switching parameters. Actually, the experiments in this research show that the new CSA algorithms with an exponentially increasing switching parameter exceeded the other variants CSA.

Also, Cheng, et al. [19] in this research A modified CS version is proposed based Dynamic Feedback Information, which is called (DFCS). The population characteristics such as fitness quality, solution enhancement levels have been The guidance information is often used to dynamically adjust the parameters of the method in terms of the feedback monitoring principle. The proposed DFCS algorithm is evaluated with various dimensions on a set of benchmark problems. The mathematical and statistical outcomes indicate that DFCS is a sustainable method compared to five new variants of CS and six state-of-the-art approaches. Wang and Zhou [20] introduced the dynamic parameter adjustment method was used to control the size of the step and the possibility of discovery.

Moreover, Rakhshani and Rahati [21] presented a new approach to make balance between global and local search ability, they implemented the training process and information sharing search approach. The suggested algorithm was evaluated by two separate datasets on 21 benchmark problems. It was shown that the enhanced version is improved the ability to explore and speeds up convergence. Mlakar, et al. [22] Suggested a modified self-adaptive CSA approach, inclusive an exploration strategy by a new balance mechanism, adaptive parameters and linear population reduction. Then, thirty benchmark issues were tested for the presented approach. The hybrid approach was found to be stronger than the standard CS and a variety of other methods. In contexts of applications, CSA was administered successfully in several areas, of example the estimation of parameters in natural systems.

Rakhshani, et al. [23], flashover voltage prediction [25], continuous dynamic optimization [24], resource optimization of datacenters [28], facility layout design [27], constrained engineering design optimization [29] and Cluster-based Wireless Senor Network Routing Issue [26],

3. CUCKOO SEARCH ALGORITHM BASED DYNAMIC PARAMETER ADJUSTMENT MECHANISM (CSDPA)

In CSA, the size of step α and Pa is an essential parameter always had to monitor the search area is identified with the practical problem. Nonetheless, these values are fixed in the initialization stage, during the process of evolution its cannot be modified in new generations. Which lead to stuck in local optima and couldn't find the best solutions.

In the earlier generations, the discovery probability value Pa and size of step α values should be sufficiently high to implement the algorithm to maximize solution vector diversity. In the final stages, these values should be reduced to outcomes in a best fine tuning of solution vectors. By the number of generation the parameters Pa and α values will be change dynamically as in equation 3 and 4,

$$\alpha(t+1) = 0.99\alpha(t) \qquad (3)$$

$$P_a = P_{a\ min} + (P_{a\ max} - P_{a\ min})\left(\frac{G_{max} - t}{G_{max}}\right) \qquad (4)$$

where *t* and *Gmax* are the current iteration and the total number of iterations, respectively. Pa min and Pa max are the minimum and maximum values of Pa, respectively. Finaly, the Lévy flight is achevied as in equation 1, when produce new solutions xi (t+1) for the ith cuckoo. Also, Using Levy flights, global exploratory random walks can be presented as in equation 2. The important steps of the CSDPA can be presented as the pseudo code based on the above principles, as shown in Figure 3.

```
Initialize the solution space dimension D
Objective function f (X), X= (f (x1, x2... xd) T
Generate initial population of n host nests
    Xi (i=1, 2... n)
set maximum iteration number Gmax
limit values of step size α;
Initialize discovery probability values
    Pa min and Pa max
While t < Max_itertions do
        Get a cuckoo randomly by Levy flights
        Evaluate its quality/ fitness Fi
        Choose a nest among n (say, j) randomly
        Update step size value α as in Eq3;
        Update discovery probability values
    Pa as in Eq4
        If Fi > Fj then
            replace j by the new solution;
        End If
        Fractions (Pa) of worse nests are
    abandoned and new
        Ones are built;
        Keep the best solutions
        Rank the solutions and find the current
    best
```

Figure 3: Pseudo code of (CSDPA)

4. RESULTS AND DISCUSSIONS

In the following experiments, a set of 10 benchmark functions will be evaluated to validate CSDPA results. Typically, these functions were used to verify optimization algorithms performance [16, 17]. Their mathematical description, search scale and global optimum are shown in Table 1 for these functions. All evaluation functions in this paper are problems of minimization. In the experiments, the parameters values with respect to CSDPA are set based on literature as in [16, 17].

The default step size values are **α** set at 0.5, the discovery probability values are set as follow $P_{a\ min}$ =0.005 and $P_{a\ max}$=1. Also, the size of population *N* is set to be 25 for all varaints of CSA, the dimension D of problems are set 10, 50 and 100 respectively, the stop criterion are representing as the maximum number of function evaluations G_{max} and it is set to 10000. Such included algorithms were simultaneously performed 30 times in each test problem to reduce stochastic error. We compared the CSDPA outcomes with standard CSA, ICS and MCSA [16, 17].

Table 1: Benchmark Optimization Function

	Name	Function	Search range	Accept
f1	Rastrigin	$f7(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[-10,10]n	0
f2	Schwefel 2.22	$f2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\Pi_{i=1}^{n}\lvert x_i\rvert$	[-100,100]n	0
f3	Schwefel 2.21	$f3(x)=max\{\lvert x_i\rvert,1\le i\le n\}$	[-100,100]	0
f4	Quartic	$f4(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	[1.28,1.28]n	0
f5	Rosenbrock	$f5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	[-100,100]n	0
f6	Sphere	$f1(x)=\sum_{i=1}^{n}x_i^2$	[-100,100]n	0
f7	Schwefel 2.26	$f6(x)=418.98288727243369*n-\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert}$	[-500,500]n	0
f8	NCRastrigin	$f8(x)=\sum_{i=1}^{n}\left[y_i^2-10\cos(2\pi y_i)+10\right]$	[-5.12,5.12]n	0
f9	Griwank	$f9(x)=\frac{1}{400}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{2}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]n	0
f10	Ackley	$f10(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	[-32,32]n	0

Table 2 shows the average (Mean) outcomes performed by CSA, ICS, MCSA, and CSDPA for the benchmark functions where D =10 and "Man" is the mean best outcomes value. As in results of Table 2, the CSDPA achieved better than the standard CSA ICS, MCSA on all functions except for *f*6 the MCSA outperform CSDPA, because the problem of stuck into local minima. In addition, table 3 represents the outcomes of CSA, ICS, MCSA, and CSDPA on the ten benchmark functions when D =50 and "Mean" is the mean best outcomes value. As in table 3, the performance of all approaches is seriously affected with the increase of dimensions. When the proposed approach is compared to CSA, ICS, MCSA, the CSDPA achieves better solutions on all benchmark functions in this experiments. In table 4, when D =100 the outcomes of 10 benchmark functions by a new approach CSDPA outperformed of other approaches reported in CSA and MCSA. In fact the CSDPA is avoiding the stuck in local optima in the most cases with deferent dimension.

Table 2: The outcomes of CSDPA and CSA variants when D = 10.

	Name	**Mean SD**	**Algorithms**			
			CSA	**ICS**	**MCSA**	**CSDPA**
f1	Rastrigin	Mean SD	5.8173E+00 2.2131E+00	3.1597E+00 9.1871E-01	8.24E-15 3.51E-14	**6.13E-32** **2.36E-29**
f2	Schwefel 2.22	Mean SD	2.5590E-04 7.0628E-04	8.0449E-10 2.3225E-09	9.93E-325 7.59E-322	**0** **0**
f3	Schwefel 2.21	Mean SD	1.19E-318 6.39E-316	NA	2.30E-325 9.84E-325	**0** **0**
f4	Quartic	Mean SD	1.14E-02 8.55E-02	NA	11.93E-03 26.69E-02	**2.83E-18** **9.15E-13**
f5	Rosenbrock	Mean SD	5.1543E+00 1.3235E+01	3.7763E+00 3.2777E+00	7.22E-33 5.64E-30	**42.06E-54** **11.2E-50**
f6	Sphere	Mean SD	2.5086E-16 3.5572E-16	4.2599E-22 1.1657E-21	**1.03E-321** **8.27E-321**	64.5E-98 48.02E-92
f7	Schwefel 2.26	Mean SD	4.7802E+02 1.8304E+02	3.8381E+02 1.6208E+02	1.49E-10 3.27E-05	**2.59E-26** **9.11E-21**
f8	NCRastrigin	Mean SD	5.8173e+00 2.2131e+00	3.1597e+00 9.1871e-01	8.24E-15 3.51E-14	**32.81E-29** **7.11E-20**
f9	Griwank	Mean SD	5.3369e-02 2.1703e-02	3.4795e-02 1.6791e-02	5.90E-16 8.25E-16	**12.4E-39** **1.22E-33**
f10	Ackley	Mean SD	1.1978e-06 2.4007e-06	1.5947e-07 6.5468e-07	4.75E-20 1.06E-19	**14.6E-42** **2.80E-36**

Table 3 The outcomes of CSDPA and CSA variants when D = 50.

	Name	Mean SD	**Algorithms**			
			CSA	**ICS**	**MCSA**	**CSDPA**
f1	Rastrigin	Mean SD	7.1486e+01 1.3445e+01	5.3111e+01 1.0093e+01	4.29E-03 9.27E-03	**7.58E-11** **12.63E-6**
f2	Schwefel 2.22	Mean SD	2.7147e+01 6.2970e+01	3.8818e-06 1.59140e-05	2.65E-29 4.99E-28	**62.17E-53** **6.14E-31**
f3	Schwefel 2.21	Mean SD	1.33E-33 1.38E-30	NA	1.28E-38 7.94E-38	**80.2E-49** **3.19E-41**
f4	Quartic	Mean SD	16.58 37.11	NA	25.86 37.58	**16.5E-17** **2.38E-9**
f5	Rosenbrock	Mean SD	1.5139e+02 1.2737e+02	7.6080e+01 3.9611e+01	8.51E-15 7.94E-15	**22.9E-28** **9.38E-16**
f6	Sphere	Mean SD	2.7691e-11 9.0314e-11	2.0675e-19 2.3478e-19	9.88E-22 7.09E-20	**51.08E-47** **12.26E-35**
f7	Schwefel 2.26	Mean SD	4.4152e+03 7.2422e+02	4.7645e+03 7.6395e+02	7.19E-02 1.60	**82.7E-13** **11.09E-8**
f8	NCRastrigin	Mean SD	7.1486e+01 1.3445e+01	5.3111e+01 1.0093e+01	4.29E-03 9.27E-03	**67.9E-18** **5.78E-12**
f9	Griwank	Mean SD	8.5133e-03 1.4217e-02	1.2358e-10 6.7400e-10	2.48E-05 9.50E-05	**43.90E-16** **2.14E-11**
f10	Ackley	Mean SD	4.0143e+00 1.0781e+00	8.0832e-01 1.2424e+00	5.08E-07 2.14E-06	**22.06E-23** **6.3E-18**

Table 4 The outcomes of CSDPA and CSA variants when D = 100.

	Name	Mean SD	Algorithms		
			CSA	**MCSA**	**CSDPA**
f1	Rastrigin	Mean SD	3.21 4.08	0.18 1.73	**29.01E-5** **17.4E-3**
f2	Schwefel 2.22	Mean SD	5.24E-08 1.07E-06	4.82E-10 7.57E-09	**8.16E-14** **6.12E-11**
f3	Schwefel 2.21	Mean SD	2.28E-10 8.86E-08	7.29E-10 4.28E-08	**7.76E-20** **8.13E-17**
f4	Quartic	Mean SD	414.98 927.70	407.5 825.42	**0.88** **1.63**
f5	Rosenbrock	Mean SD	2.78E-04 6.10E-02	2.72E-07 6.10E-05	**15.7E-16** **9.23E-13**
f6	Sphere	Mean SD	3.01E-05 9.44E-05	4.08E-09 1.05E-08	**25.19E-36** **13.45E-29**
f7	Schwefel 2.26	Mean SD	1.02 3.03	2.16 2.37	**18.3E-5** **6.08E-2**
f8	NCRastrigin	Mean SD	3.21 4.08	0.18 1.73	**19.37E-4** **5.02E-2**
f9	Griwank	Mean SD	2.14 2.96	0.05 0.73	**6.84E-7** **18.6E-4**
f10	Ackley	Mean SD	0.02 0.63	1.92E-02 2.39E-02	**5.11E-16** **4.89E-12**

5. CONCLUSIONS

In this research presented a new cuckoo search algorithm based dynamic parameter adjustment mechanism called (CSDPA). The CSDPA introduced two equation to update the parameters values of steps size and discovery probability during search process. The experiments are carried out on ten benchmark functions. The outcomes show CSDPA outperforms of the standard CSA and other CSA variant. To examine the balance among exploitation and exploration in the future to improve the CSA algorithm's ability to perform more complex issues, especially in some real-life applications.

6. REFERENCES

[1] W. Alomoush, K. Omar, A. Alrosan, Y. M. Alomari, D. Albashish, and A. Almomani, "Firefly photinus search algorithm," Journal of King Saud University - Computer and Information Sciences, 2018/07/18/ 2018.

[2] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Z. Zamli, W. Alomoush, and M. I. Younis, "Modified Opposition Based Learning to Improve Harmony Search Variants Exploration," in International Conference of Reliable Information and Communication Technology, 2019, pp. 279-287.

[3] I. Fister Jr, M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: perspectives and research challenges," Applied Mathematics and Computation, vol. 252, pp. 155-165, 2015.

[4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE computational intelligence magazine, vol. 1, pp. 28-39, 2006.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, pp. 1942-1948 vol.4.

[6] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), ed: Springer, 2010, pp. 65-74.

[7] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," arXiv preprint arXiv:1003.1409, 2010.

[8] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp. 210-214.

[9] W. Alomoush, S. N. H. S. Abdullah, S. Sahran, and R. I. Hussain, "MRI BRAIN SEGMENTATION VIA HYBRID FIREFLY SEARCH ALGORITHM," Journal of Theoretical & Applied Information Technology, vol. 61, 2014.

[10] W. Alomoush and K. Omar, "Dynamic Fuzzy C-Mean Based Firefly Photinus Search Algorithm for MRI Brain Tumor Image Segmentation," PhD,

Computer science, Universiti Kebangsaan Malaysia, Malaysia, 2015.

[11] W. K. Alomoush, S. Abdullah, S. Sahran, and R. I. Hussain, "Segmentation of MRI brain images using FCM improved by firefly algorithms," Journal of Applied Sciences, vol. 14, pp. 66-71, 2014.

[12] A. Alrosan, N. Norwawi, W. Ismail, and W. Alomoush, "Artificial bee colony based fuzzy clustering algorithms for MRI image segmentation," in International Conference on Advances in Computer Science and Electronics Engineering—CSEE, 2014, pp. 225-228.

[13] W. Alomoush, A. Alrosan, N. Norwawi, Y. M. Alomari, D. Albashish, A. Almomani, et al., "A Survey: Challenges of Image Segmentation Based Fuzzy C-Means Clustering Algorithm," Journal of Theoretical & Applied Information Technology, vol. 96, 2018.

[14] W. Alomoush and A. Alrosan, "Metaheuristic Search-Based Fuzzy Clustering Algorithms," arXiv preprint arXiv:1802.08729, 2018.

[15] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," Energy, vol. 60, pp. 99-108, 2013.

[16] M. Shehab, A. T. Khader, and M. Laouchedi, "Modified cuckoo search algorithm for solving global optimization problems," in International conference of reliable information and communication technology, 2017, pp. 561-570.

[17] E. Valian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for global optimization," International Journal of Communications and Information Technology, vol. 1, pp. 31-44, 2011.

[18] M. Mareli and B. Twala, "An adaptive Cuckoo search algorithm for optimisation," Applied computing and informatics, vol. 14, pp. 107-115, 2018.

[19] J. Cheng, L. Wang, Q. Jiang, Z. Cao, and Y. Xiong, "Cuckoo search algorithm with dynamic feedback information," Future Generation Computer Systems, vol. 89, pp. 317-334, 2018.

[20] J. Wang and B. Zhou, "A hybrid adaptive cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation," Neural Computing and Applications, vol. 27, pp. 1511-1517, 2016.

[21] H. Rakhshani and A. Rahati, "Snap-drift cuckoo search: A novel cuckoo search optimization algorithm," Applied Soft Computing, vol. 52, pp. 771-794, 2017.

[22] U. Mlakar, I. Fister Jr, and I. Fister, "Hybrid self-adaptive cuckoo search for global optimization," Swarm and Evolutionary Computation, vol. 29, pp. 47-72, 2016.

[23] H. Rakhshani, E. Dehghanian, and A. Rahati, "Hierarchy cuckoo search algorithm for parameter estimation in biological systems," Chemometrics and Intelligent Laboratory Systems, vol. 159, pp. 97-107, 2016.

[24] J. K. Kordestani, H. A. Firouzjaee, and M. R. Meybodi, "An adaptive bi-flight cuckoo search with variable nests for continuous dynamic optimization problems," Applied Intelligence, vol. 48, pp. 97-117, 2018.

[25] J. Cheng, L. Wang, and Y. Xiong, "Modified cuckoo search algorithm and the prediction of flashover voltage of insulators," Neural Computing and Applications, vol. 30, pp. 355-370, 2018.

[26] G. P. Gupta and S. Jha, "Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques," Engineering Applications of Artificial Intelligence, vol. 68, pp. 101-109, 2018.

[27] S. Kang, M. Kim, and J. Chae, "A closed loop based facility layout design using a cuckoo search algorithm," Expert Systems with Applications, vol. 93, pp. 322-335, 2018.

[28] S. M. Sait, A. Bala, and A. H. El-Maleh, "Cuckoo search based resource optimization of datacenters," Applied Intelligence, vol. 44, pp. 489-506, 2016.

[29] G. Kanagaraj, S. Ponnambalam, N. Jawahar, and J. M. Nilakantan, "An effective hybrid cuckoo search and genetic algorithm for constrained engineering design optimization," Engineering Optimization, vol. 46, pp. 1331-1351, 2014.