International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

A modified cuckoo search using different search strategies

Hüseyin HAKLI*[1]

Accepted 3rd September 2016

Abstract:Cuckoo search (CS) is one of the recent population-based algorithms used for solving continuous optimization problems. The most known problem for optimization techniques is balancing between exploration and exploitation. CS uses two search strategies to updating the nest: local and global search. Although cuckoo search are adequate for the exploration, it is not well enough the exploitation. Only one search equation is used for local search, this equation remains incapable and causes some deficiencies about the exploitation. To enhance the ability of exploitation and to balance between global search and local search, different search strategies were implemented in CS algorithm. The proposed method was compared with basic CS on well-known unimodal and multimodal benchmark functions. Experimental results show that the proposed method is more successful than the basic CS in terms of solution quality.

Keywords:*Cuckoo search, Continuous optimization, Search strategies*

1. Introduction

In the recent years, many novel nature inspired algorithm have been proposed to solve continuous optimization problems, alongside mostly known swarm intelligence algorithms such as particle swarm optimization [1], artificial bee colony [2], ant colony optimization [3] etc. The recent nature inspired algorithm was inspired by behaviours of animals in the nature. To illustrate, the firefly algorithm [4] was developed by flashing characteristic of fireflies, the bat algorithm [5] was proposed by being inspired behaviour of echolocation of bats. Yang and Deb [6] proposed the cuckoo search by being inspired breeding behaviour of some cuckoo species.

Cuckoo search (CS) is based on the obligate brood parasitism of some cuckoo species in combination with Levy flight random walk. Cuckoo search has been very popular in a short time and started to apply many engineering fields and optimization with promising efficiency and few control parameter settings. Due to these advantages, CS was used for clustering of web search results [7], hydrothermal scheduling [8,9], economic dispatch [10], multilevel thresholding [11], parameter estimation and optimization [12, 13] , redundancy allocation problems [14], forecasting solar radiation [15] and so on.

The main problem of the optimization techniques is balancing between exploration and exploitation. The exploration is concerned the ability of autonomously seeking for the global optimum, whereas the exploitation is related to the ability of applying the existing knowledge to look for better solutions [16]. In other words, exploration represents global search and the exploitation stands for local search. CS has two search techniques and uses Levy flight rather than standard random walks for the global search. Thus, CS can explore the search space efficiently [17]. Although CS has local and global search capabilities, it opens to improvement about balancing between exploitation and exploration. In addition, due to some deficiencies about exploitation, strengthening the local search is more important for the CS. Many studies were performed to improve performance of cuckoo search. The cuckoo search parameters were properly tuned by Valian et al. [18] to enhance accuracy and convergence rate of the cuckoo search. Walton et al. [19] proposed modified cuckoo search which involves the addition of information exchange between the top eggs, or best solution. To improve refining ability and convergence rate of cuckoo search, Zhang and Chen [20] proposed the cuckoo search with adaptive method. This adaptive method was used to control the scaling factor and find probability to enhance the population diversity. A new cuckoo search algorithm based on the idea of opposition (OCS) algorithm is proposed by Zhao and Li [21] to increase the exploration efficiency of solution space. They merged the opposition-based learning in the CS and the proposed algorithm fully used beneficial information of the best solutions. In order to balance the exploitation and exploration of the cuckoo search algorithm, Li and Yin [22] proposed a new approach which uses two new mutation rules based on the rand and best individuals among the entire population. In addition, this new rules were combined through a linear decreasing probability rule. CS was used in hybridization with the other algorithm such as fuzzy c means [23], NEH heuristic algorithm [24].

To improve and enhance the performance of cuckoo search, we focus on ability of exploitation and some deficiencies of local search. CS uses only one search equation for the local search and this equation remains incapable and causes some deficiencies about the exploitation. To overcome this problem and balance between global and local search, different search strategies were implemented in the CS. Modified cuckoo search with the different search strategies (CSDSS) were compared with the basic CS on twelve benchmark functions. Experimental results show that the proposed method was more successful than the basic CS algorithm for both unimodal and multimodal functions. By virtue of different search strategies, the CSDSS enhanced the efficiency of local search and improved the balance between exploration and exploitation.

The rest of the paper is divided as follows. In Section 2, the basic CS algorithm is presented. The proposed algorithm is detailed in Section

[1] *Department of Computer Engineering,Selcuk University, Turkey*
* *Corresponding Author:Email: hhakli@selcuk.edu.tr*
Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

3. Section 4 gives the experimental results and comparison of the methods. As a final, the paper is concluded with the future works.

2. Cuckoo Search

Cuckoo search was inspired by the interesting breeding behaviour such as brood parasitism of certain species of cuckoos by laying their eggs in the nests of host birds [6]. In the CS algorithm, there are mainly three principle rules as follows [6, 17]:

– Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
– The best nests with high-quality eggs will be carried over to the next generations;
– The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa ϵ (0, 1). In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

For convenience, this last rules can be approximated by the fraction pa of the n nests are replaced by new nests (with new random solutions) [6]. Each egg in a nest stands for a solution and each cuckoo can lay only one egg. For the simplest approach of algorithm, each nest has only one egg. Thus, there is no difference between egg, nest and cuckoo. All of them represent one solution.
The CS uses two search strategies and two solution search equation to generate new solutions. One of them is local random walk and other one is global random walk. The global random walk is performed by the using Levy flights [17]

$$x_i^{t+1} = x_i^t + \alpha L(s,\lambda), \tag{1}$$

$$L(s,\lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}},\ \ (s \gg s_0 > 0) \tag{2}$$

where α > 0 is the step size of scaling factor and it is generally determined with α=O(L/10), where L is characteristic scale of the problem of interest. On the other hand, the local random walk can be written as [17]:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t) \tag{3}$$

where x_j^t and x_k^t are two different solution selected by randomly, H(u) is Heaviside function, ε represents random number drawn from a uniform distribution, pa stands for switching parameter of controlling the balance between local and global random walk and s is the step size. The new nests are generated by the both of Eq.(1) and Eq.(3) in the every iteration. Detailed information about the cuckoo search, please refer to [6, 17].

3. Modified Cuckoo Search with Different Search Strategies (CSDSS)

The CS can explore the search space efficiently thanks to Levy flights random walk. But one solution search equation is used for the local search and the best solution of nest is ignored by this equation. This causes lack of the exploitation. To strengthen and increase the efficiency of local search without losing the ability of exploration, different search strategies are implemented in the local random walk phase of the cuckoo search.

$$x_i^{t+1} = x_i^t + rand \times (x_r^t - x_k^t)\ \ i = 1,2,...,N, i \neq k \neq r \tag{4}$$

$$x_i^{t+1} = x_r^t + \Phi \times (x_i^t - x_k^t)\ \ i = 1,2,...,N, i \neq k \neq r \tag{5}$$

$$x_i^{t+1} = x_i^t + \Phi \times (x_i^t - x_{best}^j)\quad i = 1,2,...,N, \tag{6}$$

$$x_i^{t+1} = x_{best}^t + \Phi \times (x_k^t - x_r^t)\ \ i = 1,2,...,N, i \neq k \neq r \tag{7}$$

Equation (4) is the same as local search update rule of the basic CS algorithm [17].In (4), (5) [25] and (7) [26], x_r^t and x_k^t are solutions randomly selected from the population at time step t, and k and r are not equal to each other and i. In (6) and (7), x_{best}^t in the equations is the best solution obtained by the population so far. Also, Φ is a random number in range of [-1,1] and produced for each cuckoo which is updated at the time step t. To increase and protect the efficiency of exploration, the random neighbours are used in Eq. (4) and Eq. (5). Also, Eq. (6) and Eq. (7) are implemented to support the local search around the global best solution of population and enhance the ability of exploitation. The important point is how these equations are used together to obtain new solutions. Usage of these equations is explained by Fig.1.

```
for i=1:number of cuckoo
   if rand < ls_rate
       if( rand < 0.5)
           Generate new cuckoo with Eq. (7)
       else
           Generate new cuckoo with Eq. (6)
       end
   else
       if( rand < 0.5)
           Generate new cuckoo with Eq. (5)
       else
           Generate new cuckoo with Eq. (4)
       end
   end
end
```

Figure1. Usage of equations for the proposed method

In Fig.1, *ls_rate* is the constant parameter in range of [0,1] that controls the level of local search for the proposed method. If the *ls_rate* is high, the proposed method searches the around of global best solution and makes more local search. The proposed method uses the algorithm in Fig.1 instead of the Eq. (3) in the basic CS to generate new solutions.

4. Experimental Results

In this study, all the experiments were run on a machine with Intel Core i7 2.40 GHz CPU, 8 GB RAM and the Windows 8 operating system and the codes were implemented in Matlab 2014 (8.3).
To examine the performance and accuracy of the CSDSS algorithm, the CS and CSDSS are applied to optimize 12 benchmark functions. The dimension is set 30 for all benchmark functions. Experiments are implemented with the population size of 40 for both algorithms. As a stopping criterion is used the maximum number of iteration and it is determined as 2500 in each run for each test function. While pa parameter is set 0.25 for basic CS algorithm, it is determined as 0 for the proposed method. In addition, *ls_rate* is determined as 0.3 for the CSDSS as a result of the trials.
The determined benchmark functions are numbered f1 to f12 given in Table 1. All the functions are to be minimized. Table 1 shows characteristic (C), search range (Range), and formulation of the functions. The cuckoos are initialized in the search range. The functions f1, f2, f3, f4, f9, f10 and f11 are unimodal and f5, f6, f7, f8 and f12 are multimodal.

Table 1. Benchmark Functions (C: Characteristic, U: Unimodal, M: Multimodal)

Range	C	Function	Formulation
[-100, 100]	U	Sphere	$f_1 = \sum_{i=1}^{n} x_i^2$
[-10, 10]	U	Schwefel2.22	$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
[-10, 10]	U	Rosenbrock	$f_3 = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$
[-1.28, 1.28]	U	QuarticWN	$f_4 = \sum_{i=1}^{n} i x_i^4 + random[0,1)$
[-5.12, 5.12]	M	Rastrigin	$f_5 = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$
[-32, 32]	M	Ackley	$f_6 = -20\exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right\} + 20 + e$
[-600, 600]	M	Griewank	$f_7 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
[-50, 50]	M	Penalized2	$f_8 = \frac{1}{10}\{\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{i+1})]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
[-100, 100]	U	Step	$f_9 = \sum_{i=1}^{n}\left(\lfloor x_i + 0.5 \rfloor\right)^2$
[-100, 100]	U	Elliptic	$f_{10} = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$
[-10, 10]	U	SumSquare	$f_{11} = \sum_{i=1}^{n} i x_i^2$
[-10, 10]	M	Alpine	$f_{12} = \sum_{i=1}^{n} \lvert x_i \cdot \sin(x_i) + 0.1 \cdot x_i \rvert$

Table 2 contains a comparison of the original CS and the proposed method for the 12 benchmark functions listed in Table 1. Table 2 shows the mean result of 30 runs with 2500 iterations and dimension 30. The information about the mean optimum solution, standard deviation, minimum and maximum value and Wilcoxon test result by CS and CSDSS in 30 runs over benchmark functions are given in Table 2. The best mean result and the best standard deviations of the benchmark functions obtained by the algorithms are shown in bold. In the Table 2, in case Wilcoxon test was at 0.05 significance level, + mark was used, otherwise, − mark was used; NA was used in case all results showed the same value or optimum value.

Table 2. Results for benchmark functions (Opt:Optimum, STD:Standard Deviation,Min:Minimum,Max:Maximum, Sign: Statistical Test Sign)

Func.	Opt.		CS	CSDSS	Sign
f_1	0	Mean	1.14E-15	**2.87E-110**	+
		STD	8.17E-16	**1.34E-109**	
		Min	1.71E-16	2.64E-129	
		Max	3.22E-15	7.33E-109	
f_2	0	Mean	2.75E-05	**1.10E-49**	+
		STD	1.59E-05	**6.04E-49**	
		Min	7.55E-06	5.53E-62	
		Max	7.39E-05	3.31E-48	
f_3	0	Mean	**2.00E+01**	2.53E+01	+
		STD	**1.69E+00**	8.21E-01	
		Min	1.60E+01	2.27E+01	
		Max	2.27E+01	2.67E+01	
f_4	0	Mean	1.42E-02	**6.84E-03**	+
		STD	5.01E-03	**9.79E-03**	
		Min	5.70E-03	1.03E-04	
		Max	2.78E-02	3.53E-02	
f_5	0	Mean	5.23E+01	**3.04E+01**	+
		STD	8.18E+00	**1.35E+01**	
		Min	3.81E+01	0.00E+00	
		Max	6.84E+01	5.48E+01	
f_6	0	Mean	2.01E-01	**3.55E-15**	+
		STD	5.23E-01	**0.00E+00**	
		Min	2.84E-05	3.55E-15	
		Max	2.33E+00	3.55E-15	
f_7	0	Mean	1.79E-05	**0.00E+00**	+
		STD	6.85E-05	**0.00E+00**	
		Min	3.28E-11	0.00E+00	
		Max	3.67E-04	0.00E+00	
f_8	0	Mean	**7.62E-11**	3.88E-01	+
		STD	**9.92E-11**	2.95E-01	
		Min	4.52E-12	1.73E-02	
		Max	4.58E-10	1.33E+00	
f_9	0	Mean	**0.00E+00**	**0.00E+00**	**NA**
		STD	**0.00E+00**	**0.00E+00**	
		Min	0.00E+00	0.00E+00	
		Max	0.00E+00	0.00E+00	
f_{10}	0	Mean	6.76E-10	**9.11E-107**	+
		STD	3.93E-10	**3.20E-106**	
		Min	1.39E-10	3.47E-123	
		Max	1.66E-09	1.43E-105	
f_{11}	0	Mean	6.14E-14	**1.04E-106**	+
		STD	4.03E-14	**5.71E-106**	
		Min	7.11E-15	2.17E-131	
		Max	1.37E-13	3.13E-105	
f_{12}	0	Mean	4.04E+00	**5.70E-02**	+
		STD	8.46E-01	**1.54E-01**	
		Min	1.90E+00	3.66E-63	
		Max	5.54E+00	8.15E-01	

When examines Table 2, CSDSS algorithm acquires better results compared to CS algorithm for the all benchmark functions except f3 and f8 in terms of solutions. The proposed method is more successful for the all unimodal functions. While ability of local search of CS algorithm remains incapable on the unimodal functions, the proposed algorithm makes more sufficient local search and obtains the better result than the CS. The CS algorithm is better than proposed method for f3 and f8 functions. But, the CSDSS gives the optimum results for the function f7. Experimental results show that not only the CSDSS enhances the ability of local search but also improves the balance between exploration and exploitation. The proposed method outperforms the basic CS on the unimodal functions, at the same time it obtains better results the most known multimodal functions such as Rastrigin, Griewank and Ackley.

Non-parametric statistical test has been implemented for statistically significant of between CS and the CSDSS. From the results in Table 2, it can be observed that the CSDSS algorithm's performance is significantly better compared to CS algorithm for all benchmark functions.

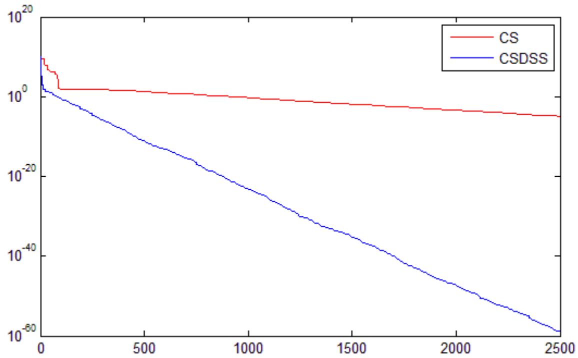

Figure 2. Convergence graph of Schwefel2.22 function

In order to compare convergence rates of the methods, convergence graphs for CS and CSDSS algorithms are given in Fig. 2, Fig. 3 and Fig. 4 for Schwefel2.22, Rastrigin and Griewank functions. The convergence graphs show that the proposed method has rapidly convergence and also obtains successful results. Examining Fig.2, while the CS converges slowly, the CSDSS continues the improving solutions thanks to ability of local search. For the Rastrigin function, the proposed method has early convergence and then gets stuck the local minima in Fig.3. Although its early convergence, it obtains better result than the basic CS.

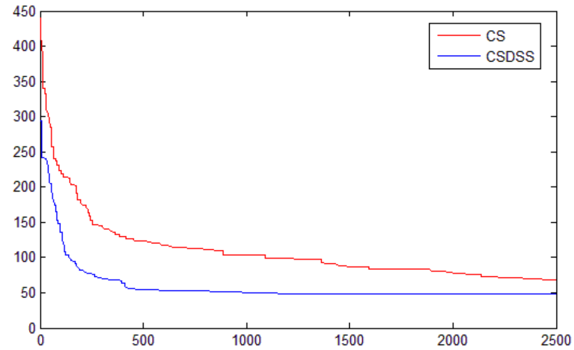

Figure 3. Convergence graph of Rastrigin function

When examines Fig.4, the CSDSS outperforms the CS algorithm and reaches the optimum result very quickly. So, with respect to overall assessment of convergence graphs, we can say that the proposed method converges quickly and accurately towards the optimum solution.

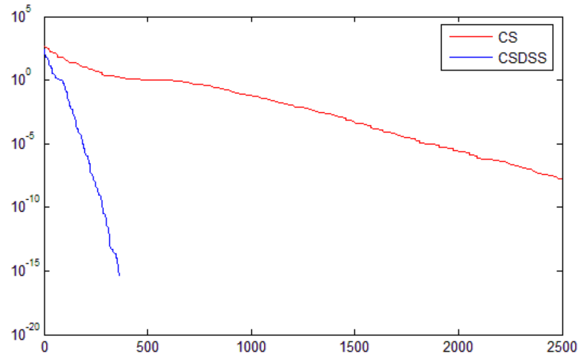

Figure 4. Convergence graph of Griewank function

5. Conclusion

In the basic CS, one solution search equation is used for the local search and the best solution of nest is ignored by this equation. This causes lack of the local search. In order to overcome these deficiencies of the CS algorithm in local search, this paper proposed the CSDSS algorithm using the different search strategies. Not only the proposed algorithm strengthens the ability of local search but also it improves the balance between exploration and exploitation. The proposed algorithm was compared with the basic CS on twelve benchmark functions. The CSDSS performs better results than the basic CS for all benchmark functions except f3 and f8. The proposed method is more efficient and more effective than the basic CS algorithm especially on the unimodal and some of multimodal functions.

As future work, the proposed method will be implemented to optimization problems and different search strategies will be used for the other nature-inspired algorithms. Furthermore, different search strategies can be increased and varied to improve the success of the proposed algorithm.

Acknowledgements

This study has been supported by No. 15611818Scientific Research Project of Selcuk University. This study has been presented as an oral presentation at ICAT'2016 conference held in Konya (Turkey), 1-3 September 2016 and selected for the journal of IJISAE (International Journal of Intelligent Systems and Applications in Engineering).

References

[1] J. Kennedy, R. Eberhart, "Particle swarm optimization", in: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[2] D. Karaboga, "An idea based on honey bee swarm for numerical optimization",Technical Report-TR06, Erciyes University, Engineering Faculty, Comput. Eng.Dep., 2005.

[3] M. Dorigo, G.D. Caro, "Ant colony optimization: a new meta-heuristic", in: *Proceedings of the 1999 Congress on Evolutionary Computation*, 1999, pp. 1470–1477.

[4] X.-S. Yang, "Firefly algorithms for multimodal optimization", in: *5th International Symposium SAGA*, 2009, pp. 169–178.

[5] X.-S. Yang, "A new metaheuristic bat-inspired algorithm", in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (Eds. Cruz C., Gonzalez J., Krasnogor N., and Terraza G.), Springer, SCI 284, 65-74, 2010.

[6] X.-S. Yang and S. Deb, "Cuckoo search via Le´vy flights", in *Proceedings of world congress on nature and biologically inspired computing IEEE Publications*, 2009, pp. 210–214.

[7] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendoza, E. León and E. Herrera-Viedma, "Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion", *Information Sciences*, vol. 281, pp. 248–264, 2014.

[8] T. T. Nguyen, D. N. Vo and T. T. Dao, "Cuckoo Search Algorithm Using Different Distributions for Short-Term Hydrothermal Scheduling with Cascaded Hydropower Plants", in *Proc.TENCON'14* , 2014, pp. 1-6.

[9] T. T. Nguyen, D. N. Vo and A. V. Truong, "Cuckoo search algorithm for short-term hydrothermal scheduling", *Applied Energy,* vol. 132 pp. 276–287, 2014.

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

[11] S. Agrawal, R. Panda, S. Bhuyan and B.K. Panigrahi, "Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm", *Swarm and Evolutionary Computation* , vol. 11, pp. 16–30, 2013.

[12] X. Ding, Z. Xu , N.J. Cheung and X. Liu, "Parameter estimation of Takagi–Sugeno fuzzy system using heterogeneous cuckoo search algorithm", *Neurocomputing*, vol. 151, pp. 1332–1342, 2015.

[13] S. Berrazouane and K. Mohammedi, "Parameter optimization via cuckoo optimization algorithm of fuzzy controller for energy management of a hybrid power system", *Energy Conversion and Management* , vol. 78, pp. 652–660, 2014.

[14] G. Kanagaraj, S.G. Ponnambalam and N. Jawahar, "A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems", *Computers & Industrial Engineering* , vol. 66, pp. 1115–1124, 2013.

[15] J. Wang, H. Jiang, Y. Wu and Y. Dong, "Forecasting solar radiation using an optimized hybrid model by Cuckoo Search algorithm", *Energy* , vol. 81, pp. 627-644, 2015.

[16] G. Li , P. Niu and X. Xiao , "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization", Applied Soft Computing, Vol. 12, pp. 320–332, 2012.

[17] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications", *Neural Comput & Applic,* vol. 24, pp. 169–174, 2014.

[18] E. Valian, S. Tavakoli, S. Mohanna and A. Haghi, "Improved cuckoo search for reliability optimization problems", *Computers & Industrial Engineering* , vol. 64, pp. 459–468, 2013.

[19] S. Walton, O. Hassan, K. Morgan and M.R. Brown, "Modified cuckoo search: A new gradient free optimisation algorithm", *Chaos, Solitons & Fractals* , vol. 44, pp. 710–718, 2011.

[20] Z. Zhang and Y. Chen, "An Improved Cuckoo Search Algorithm with Adaptive Method", in: *Proceedings of 2014 Seventh International Joint Conference on Computational Sciences and Optimization,* 2014, pp. 204-207.

[21] P. Zhao and H. Li, "Opposition-Based Cuckoo Search Algorithm for Optimization Problems", in: *Proceedings of 2012 Fifth International Symposium on Computational Intelligence and Design,* 2012, pp. 344-347.

[22] X. Li and M. Yin, "Modified cuckoo search algorithm with self adaptive parameter method", *Information Sciences* , vol. 298, pp. 80–97, 2015.

[23] D. Binu , M. Selvi and A. Georgea, "MKF-Cuckoo: Hybridization of Cuckoo Search and Multiple Kernel-Based Fuzzy C-Means Algorithm", *AASRI Procedia* , vol. 4, pp. 243 – 249, 2013.

[24] M.K. Marichelvam, T. Prabaharan and X.S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", *Applied Soft Computing* , vol. 19, pp. 93–101, 2014.

[25] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE T. Syst. Man cy. B,, doi: 10.1109/TSMCB.2012.2222373, 2012.

[26] W. Gao, S. Liu, L. Huang, A global best artificial bee colony algorithm for global optimization, J. Comput. Appl. Math. 236 (2012) 2741-2753.