Covenant Journal of Informatics & Communication Technology. Vol. 7 No. 2, Dec. 2019
ISSN: p2354 – 3566 e 2354 – 3507 DOI:
An Open Access Journal Available Online

Development of a Dynamic Cuckoo Search Algorithm

Sanusi, Yusuf Audee[1], Muhammed, Bashir Mu'azu[1], Sani, Man-Yahaya[2], Zaharuddeen, Haruna[1], Salawudeen, Ahmed Tijani[1] & Prosper Oyibo[1]

[1] Ahmadu Bello University, Zaria, Nigeria
[2] Federal Polytechnic, Bida, Nigeria
Contact(s). audeesy@gmail.com, muazumb1@yahoo.com, sanimanyahaya@gmail.com, elzet2007@gmail.com, elteejayy@gmail.com, prospy_711@yahoo.com

Date Received- 18.10.2018 Accepted- 20.08.2019
Date of Publication: December, 2019

Abstract— This research is aimed at the developing a modified cuckoo search algorithm called dynamic cuckoo search algorithm (dCSA). The standard cuckoo search algorithm is a metaheuristics search algorithm that mimic the behavior of brood parasitism of some cuckoo species and Levy flight behavior of some fruit flies and birds. It, however uses fixed value for control parameters (control probability and step size) and this method have drawbacks with respect to quality of the solutions and number of iterations to obtain optimal solution. Therefore, the dCSA is developed to address these problems in the CSA by introducing random inertia weight strategy to the control parameters so as to make the control parameters dynamic with respect to the proximity of a cuckoo to the optimal solution. The developed dCSA was compared with CSA using ten benchmark test functions. The results obtained indicated the superiority of dCSA over CSA by generating a near global optimal result for 9 out of the ten benchmark test functions.

Keywords/Index Terms— cuckoo search algorithm, control parameters, dynamic cuckoo search algorithm, global optimal solution, inertia weight strategy.

1. Introduction

Nature inspired optimization techniques have been proven in solving many optimization problems efficiently (Yang, 2012). Optimization is a process of producing solutions to problem subjected under constrained situations by utilizing the resources available in the best possible way (Yılmaz & Küçüksille, 2015). Nature inspired metaheuristic algorithms forms a significant part of modern soft computing, computational intelligence and global optimization algorithms. Optimization algorithms are of two categories: deterministic and stochastic (Yang, 2010a). Deterministic algorithms are search algorithms that generate same result as long initial conditions does not change. Stochastic algorithms are search algorithms that uses randomness in their search process there by generating different solutions at each run even when initial conditions does not change (Yılmaz & Küçüksille, 2015). Heuristic and metaheuristic approaches are the two types of stochastic algorithm. Heuristic methods are problem dependent methods where each technique can only be used for a single kind of optimization problem. The metaheuristic - based search algorithm is a general solver algorithm that can be used for solving different kinds of optimization problems (Shehab *et al.*, 2017). Metaheuristic search algorithms use exploration (diversification) and exploitation (intensification) to generate a global solution. Exploration process guides the algorithm to search for best local solutions within the solution search space. Exploitation process guides the algorithm to search for global optimum solution within the generated local solutions. A balance between exploration and exploitation enables the metaheuristic search algorithm to converge to the global optimum solution (Civicioglu & Besdok, 2013).

The power of metaheuristic search algorithms comes from their inspiration of nature, especially biological systems. These nature inspired metaheuristic search algorithms have been widely applied in solving optimization problems (Yang & Deb, 2009). Example of nature inspired metaheuristic search algorithms are: particles swarm optimization was inspired by fish schooling and swarm of birds (Eberhart & Kennedy, 1995), firefly algorithm mimic the flashing pattern of fireflies (Yang, 2009), cat swarm optimization algorithm mimic the trace and catch behavior of cats against their prey (Chu *et al.*, 2006), bat algorithm works mimic the echolocation behavior of micro bats (Yang, 2010b), ant colony optimization algorithm mimic the ant foraging behavior in their colonies (Dorigo & Thomas, 2004), etc.

Cuckoo search algorithm (CSA) which is also a nature inspired metaheuristic search algorithm was developed by Yang and Deb (2009). The algorithm mimics the brood parasitic behavior of some cuckoo species and Levy flight behavior of some fruit flies and birds. CSA has been proved to be an effective optimization algorithm when compared with other algorithms (Vaijayanthi *et al.*). The parameters; switching probability $\left(P_a\right)$ and step size $\left(\alpha\right)$ used in CSA respectively guide the algorithm to generate improved solutions globally and locally. These parameters are significance in fine tuning of solutions and are utilized in the adjustment of convergence speed of the algorithm (Valian *et al.*, 2011). The algorithm has been applied in obtaining optimal features, obtaining optimized parameters of several classifiers including artificial neural network (ANN), support vector machines (SVM) parameters, etc. (Kamat & Karegowda, 2014). The algorithm is simple and effective, it has been successfully applied to real time optimization problems (Li *et al.*, 2014).

The drawbacks of fixed value of P_a and α used in the algorithm however, affect the generation of an optimal solution by increasing the convergence time and decreasing the quality of the solution (Valian *et al.*, 2011). Based on the tuning of these fixed parameters, it is discovered that a large value of P_a and small value of α increases the speed of convergence but decreases the quality of solution. Whereas, a small value of P_a and a large value of α increases the quality of solution but decreases the performance and convergence speed of the algorithm (Zhang *et al.*, 2016). Likewise, the standard CSA update the current solution by using Levy flight which is based on methods of Markov chain, to generate the global solution based on the current solution and the transition probability. This search process of the algorithm slow down the convergence speed and lower its accuracy (Qu & He, 2015).

Thus, to improve the convergence speed of the standard CSA and avoid its possibility of converging to local minima, several variations of the CSA have been developed (Li *et al.*, 2014). This research also focuses on the modification of the standard CSA by introducing inertia weight to the control parameters, P_a and α. The inertia weight will dynamically define the control parameters with respect to the position of a cuckoo in the solution search space and established balance between exploration and exploitation. The proposed modified CSA with dynamic control parameter is aimed at improving the exploitation capability and increase the convergence speed of the standard CSA.

2. Cuckoo Search Algorithm

This algorithm belongs to the class of swarm intelligence algorithm that is inspired by the strategy of cuckoo bird reproduction in combination with the behavior of Lèvy flight of some fruit flies and birds. Cuckoo birds lay their eggs in a randomly chosen nest of some birds by removing host eggs thereby increasing the chance of hatching their own laid eggs (El Aziz & Hassanien, 2016). Yang and Deb in 2009 developed CSA by mimicking the brood parasitism of some cuckoo species (Fister Jr *et al.*, 2013), in conjunction with the behavior of Lèvy flight of some fruit flies and birds (Yang & Deb, 2009).

The CSA has been summarized into three idealized rules (Yang & Deb, 2009):

a. Each cuckoo lays an egg in a randomly selected nest at a time;
b. The nests with eggs eggs will be passed over to next generations;
c. The probability of discovering an alien egg by the host bird is define by $P_a \in [0,1]$. Thus, the discovered egg can either be throw away or the nest is abandoned so as to build a new nest completely.

For simplicity and implementation of the last rule, a fraction P_a (called switching probability) of the nest population are replaced by a randomly generated nests as new solutions (Yang & Deb, 2009). In fact, this parameter establish the balance between exploration and exploitation of the CSA search process (Fister Jr *et al.*, 2013).
The local search stage of the algorithm employs a balanced combination of random walk (local and global explorative random walk) controlled by the P_a as switching parameter. Equation (1) present the mathematical implementation of local random walk (Yang & Deb, 2014):

$$x_i^{t+1} = x_i^t + \alpha s \otimes H\left(P_a - \in\right) \otimes \left(x_j^t - x_k^t\right) \quad (1)$$

.

Where; x_j^t and x_k^t denotes two different solutions randomly selected by random permutation, s denotes step size, H(u) is a Heaviside function define as a unit step discontinuous function whose value is zero and one for negative and positive argument respectively (Weisstein, 2002), $\in$ denotes a random number selected from a uniform distribution.
The global random walk for exploring the solution search space utilized a Lévy flights to generate new solutions. This is mathematically modelled as equation (2) (Yang & Deb, 2014):

$$x_i^{t+1} = x_i^t + \alpha \oplus L\left(s, \lambda\right) \quad (2)$$

Where $L(s, \lambda)$ is obtained using equation (3)

$$L(s,\lambda) = \frac{\lambda \Gamma(\lambda) \sin\left(\pi \frac{\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s >> s_0 > 0) \quad (3)$$

In equation (2), new solutions (x_i^{t+1}) are generated when for a cuckoo *i* using a Lévy flight with a step size α > 0 modelled as the scales of the problem of interests. $\oplus$ denotes an entry - wise multiplication similar to the one used in PSO, but the random walk based Lévy flight here explore the solution search space more efficiently in the long run when step length is much longer (Fister Jr *et al.*, 2013; Yang & Deb, 2009).

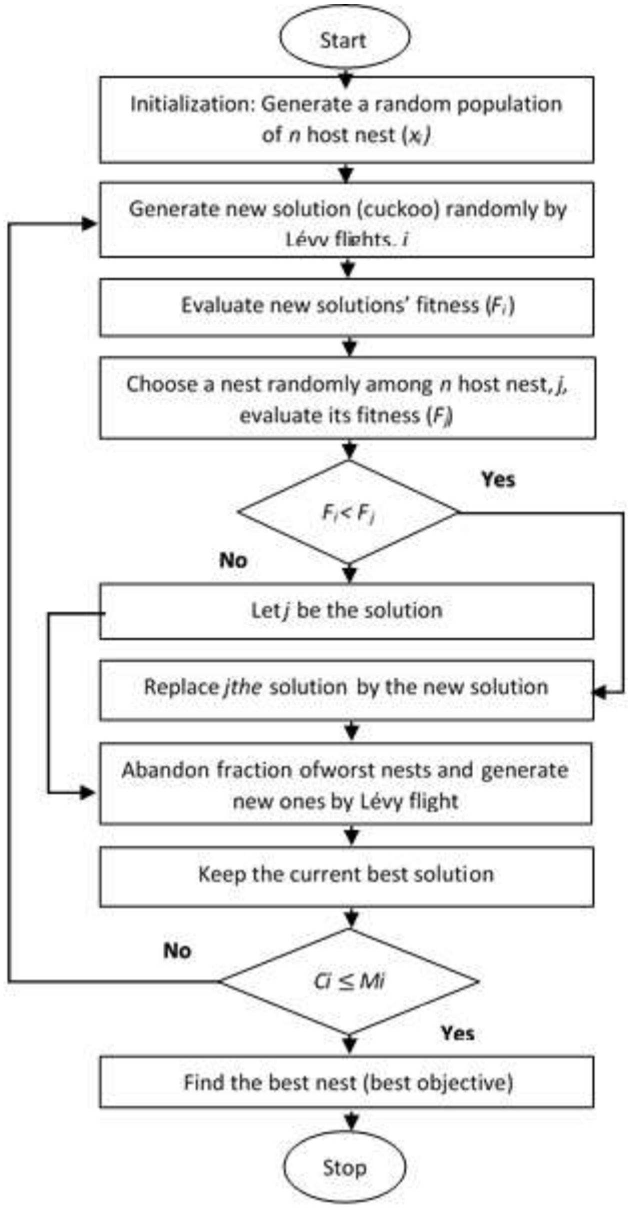

Figure 1. Flow Chart of Standard CSA

3. Dynamic Cuckoo Search Algorithm

The development of dynamic CSA was built upon the existing standard CSA, dynamic control parameters was developed by incorporating random inertia weight strategy to the control parameters of the CSA so as to improve the convergence speed and accuracy of the standard CSA.

3.1 Random Inertia Weight Strategy

The concept on inertia weight was first introduced in 1998 by Shi and Ebahart for the purpose of tuning the parameters of PSO algorithm (Bansal *et al.*, 2011). According to Chauhan *et al.* (2013), inertia weight is a function of evolution speed and aggregation degree factor that dynamically changes based on evolution. Large inertia weight is use for global search while for local search, a small inertia weight is used (Shi & Eberhart, 1998). Inertia weight approaches include linear, nonlinear, exponential, adaptive or self - adaptive, distribution based random adjustments, chaotic and fuzzy rules based strategies (Chauhan *et al.*, 2013). Inertia weight strategy regulates the trade - off between global and local search of a swarm based algorithm. The balanced between global and local search of the algorithm increases the convergence speed (Ojha & Das, 2012).

In order to enhance the exploitation capability of CSA, the idea of inertia weight was introduced to the control parameters of CSA in the form of dynamic value of iteration weight and was implemented.

The dynamic value iteration weight given in equation (4) was introduced in order to improve the convergence speed and optimal performance of the standard CSA.

$$w = 0.5 + \frac{rand}{2} \quad (4)$$

Based on equation (3.1), the fixed control parameters of the standard CSA were made to be dynamic with respect to the position of the cuckoo as the iteration increases. The resulting control parameters i.e. control probability and step size are respectively modified as shown in equation (5) and (6).

$$pa = 0.25 \times (0.5 + \frac{rand}{2}) \quad (5)$$

$$\alpha = w \times step \times (s - best) \quad (6)$$

where *rand* is a uniform random number, α is the step size, *w* is the inertia weight *s* is a randomly chosen nest and *best* is the current best solution.

The local search equation of the modified CSA is then written in equation (7)

$$x_i^{t+1} = x_i^t + \alpha w \oplus \mathcal{L}(s, \lambda) \quad (7)$$

Where the step size (α) controls the heavy tailed step size in generating new solutions.

3.2 Benchmark Test Functions

The test of efficiency, validation and reliability of optimization search algorithms has been carried in literature set of selected benchmarks test functions (Jamil & Yang, 2013). There are different benchmark test functions that are used for testing the performance of new and modified optimization algorithms (Yang & Deb, 2009). Ten of such optimization test functions used in this research work.

To enhance the understanding of these benchmark test functions by visualizing the local minimal point, Figure 2 – 11 were generated from MATLAB environment to show the shapes and surfaces of the test functions (Haruna *et al.*, 2017).

a. *Ackleys' function*

This is one of the classical functions used in testing several continuous optimization techniques. It has a single global minimum surrounded by many local minima. Equation (8) presents the mathematical model of this function (Li *et al.*, 2013):

$$f_{(x)} = -20\exp\left[-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right] + 20 + e$$

(8)

Where $n = 1, 2, \ldots$ and its' test area is typically limited to hyphercube $-32.768 \le x_i \le 32.768$ for i=1,2,…,n. The global minimum of the function is $f_{(x)} = 0, at\ x_* = (0,0, \ldots \ldots, 0)$.

The Ackley function visualization in 3D is as shown in Figure 2

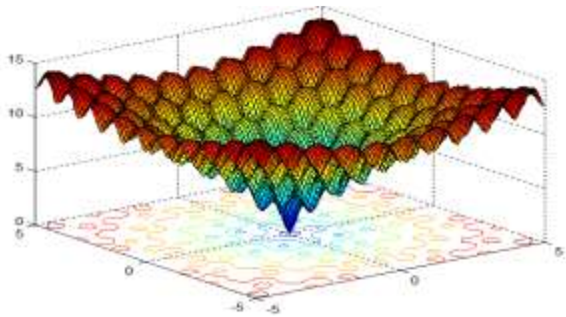

Figure 2. Ackley Function

b. De Jong's first function

This function is among the simplest benchmark function, which is continuous, unimodal and convex. Its mathematical expression is shown in equation (9) (Molga & Smutnicki, 2005):

$$f_{(x)} = \sum_{i=1}^{n} x_i^2 \qquad (9)$$

The search boundary of this function is $-5.12 \le x_i \le 5.12, i = 1, \ldots. n.$ with global minimum at `$f_{(x)} = 0, for\ x_i = 0, i = 1, \ldots, n.$ The visualization of the function in 3D is presented in Figure 3.

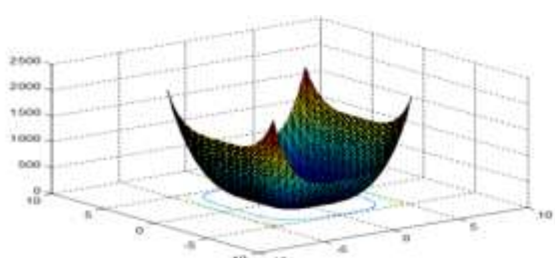

Figure 3. De Jong Function

c. Easom's function

This is another unimodal function, with global minimum surrounded with many local minima and it has small area comparative to the solution search space. This function is a minimization problem with only two variables. Equation (10) presents the mathematical description of the function (Molga & Smutnicki, 2005):

$$f_{(x,y)} = -\cos(x)\cos(y)\exp\left(-(x-\pi)^2-(y-\pi)^2\right) \quad (10)$$

The global minimum of the function is $f_{(x)} = -1$ at (π, π) in a minute region. The 3D visualisation of Easom function is as shown in Figure 4.

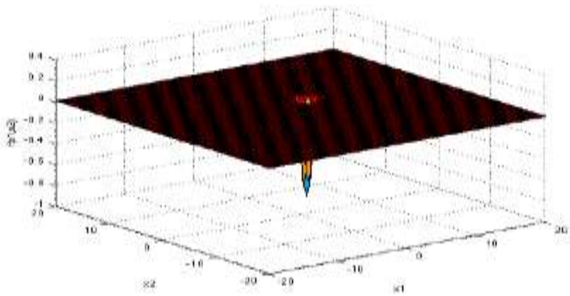

Figure 4. Easom Function

d. Griewangk's function

Griewangk's function contains many widespread local minima regularly distributed, but a single global minimum, the function is expressed mathematically as (Yang & Deb, 2009):

$$f_{(x)} = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (11)$$

The global minimum of this function is $f_{(x)}=0$, at $x_i=0$, for $i=1,\ldots n$. The 3D visualization of this function is shown in Figure 5.

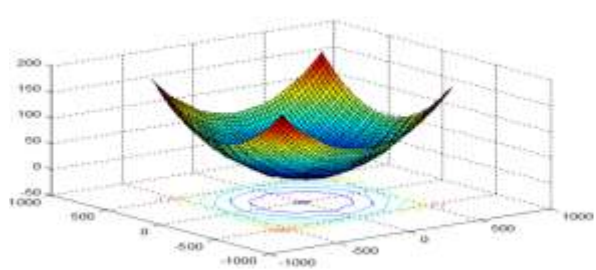

Figure 5. Greiwangk Function

e. Michalewicz's function

This function belongs to the class of multimodal test function with ***d!*** local optima. The "steepness" of the valleys or edges is defined by a parameter 'm'. The more the size of m the more difficult the search become. When the size of m is very large, the function acts like a needle in haystack (i.e. the values outside the narrow peaks of points in the solution search space gives slight information on the position of the global optimum). The function is mathematically expressed as (Molga & Smutnicki, 2005):

$$f_{(x)} = -\sum_{i=1}^{d}\sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{2m}$$

(12)

Where $m=10$, $0 \le x_i \le \pi$ and $i=1,2,...,d$. The global minimum is $f_{(x)} \approx -1.801$ for $d=2$, while $f_{(x)} \approx -4.6877$ for $d=5$.

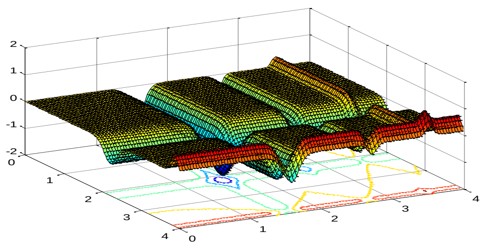

Figure 6. Michalewicz's function

f. Rastrigin's function

This function is derived from De Jong's function by adding cosine modulation to generate regular local minima. It is highly multimodal but the locations of the minima are frequently distributed. Equation (13) defined the mathematical model of this function as (Molga & Smutnicki, 2005):

$$f_{(x)} = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right] \quad (13)$$

The global minimum of this function is $f_{(x)} = 0, at\ (0,0,\dots,0)\ for -5.12 \le x_i \le 5.12,$ where $i=1,2,...,d$. The 3D visualization of this function is presented in Figure 7.

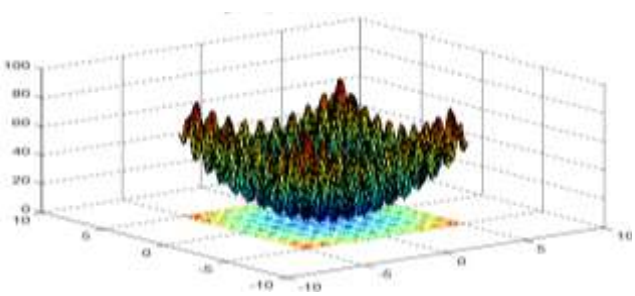

Figure 7. Rastrigin Function

g. Rosenbrock's function

The valley of this function (also called Banana function) is considered as classic optimization problem (Tang *et al.*, 2007). The function has a global optimum inside flat valley that is long, narrow and parabolic shaped. It is trivial obtaining the valley but difficult to converge to global optimum. This function has been used in determining the performance of many optimization search algorithms. The mathematical expression of this function is presented in equation (14) (Yang, 2010b):

$$f(x)=\sum_{i=1}^{d-1}((x_i-1)^2+100(x_{i+1}-x_i^2)^2)$$
(14)

The global minimum of this function is $f_{(x)}=0$ occurs at $x_i=(1,1,...,1)$ in the search domain of $-2.048\le x_i\le 2.048$ where $i=1,2,...,d-1$. The 2D representation of equation (14) is shown in equation (15)

$$f(x,y)=(x-y)^2+100(y-x^2)^2 \qquad (15)$$

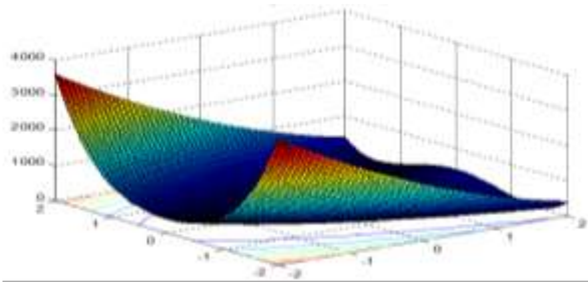

Figure 8. Rosenbrock Function

h. Schwefel's function

Schwefel's function is a multimodal function with global minimum geometrically distant over the solution search space from the successive best local minima. This function is defined in equation (16) (Yang & Deb, 2009);

$$f_{(x)}=\sum_{i=1}^{d}\left[-\sin\left(\sqrt{|x_i|}\right)\right] \qquad (16)$$

Test areas is usually bounded to hypercube $-500\le x_i\le 500, for\ i=1,...,d$. The global minimum of this function is $f_{(x)}=-418.9829$ at $x_i=420.9687,\ i=1,...,d$. The global point of this function is at 0, which is shown in the 3D visualization of Figure 9.

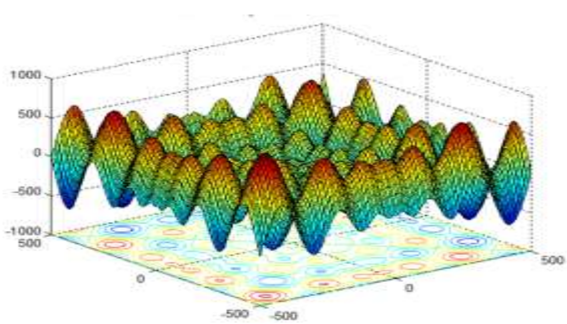

Figure 9. Schwefel Function

i. Shubert's bivariate function

This is also a multimodal function with only two variables and mathematically expressed in equation (17) (Yang & Deb, 2009);

$$f_{(x,y)} = -\sum_{i=1}^{5} i\cos\left[(i+1)x+1\right]\sum_{i=1}^{5}\cos\left[(i+1)y+1\right] \quad (17)$$

It has 18 local minima in the area $(x,y)\in[-10,10]\times[-10,10]$. The global minima of this function is $f_{(x)} = -186.7309$. Figure 10 shows a 3D visualization of Shubert function.

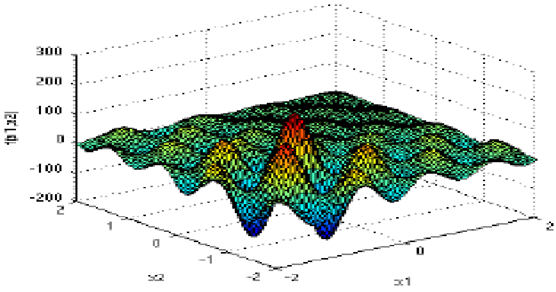

Figure 10. Shubert Function

j. Sphere function

This is a De Jong function in its simplest form. Sphere function is a unimodal and convex function mathematically represented in equation (18) (Yang, 2010b)

$$f(x) = \sum_{i=1}^{d} x_i^2 \quad (18)$$

The local minimum of this function is $f_* = 0$ at $x_* = (0,0,...,0)$ in a boundary of $-15 \le x_i \le 15$. The 3D visualization of sphere function is shown in Figure 11.

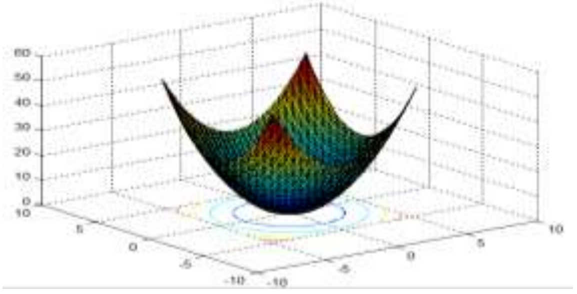

Figure 11: Sphere Function

4. Simulation Results

The developed algorithm has been implemented and coded in MATLAB (R2013b) environment and tested on an Intel Core i3–2350M, 2.30GHz with RAM of 4GB. The dynamic dCSA algorithm control parameters are defined as: population size 15, control probability range 0 – 0.25, step size range 0.1 – 1 and runs 25 times. The control parameters of the standard CSA algorithm are: control probability = 0.25, step size = 1.

The results obtained both for the developed dCSA and the standard CSA algorithm, are tabulated in Table I. Table I represents performance of the algorithms for optimizing ten optimization benchmark test functions with respect to their global optimal. Both algorithms were evaluated using the optimization benchmark test functions for 25 runs each.

From the results in Table I, it is obvious that dCSA outperforms the CSA with respect to the global optimal value in almost all the optimization benchmark functions except in test case 6 (Rastrigin function) where CSA outperformed the dCSA. However, CSA did well for this class of functions (low-dimensional and relatively easy functions), but perform fairly on others (high-dimensional and more complex functions). The superiority of dCSA over CSA is expected as inertia weight factor was incorporated into the control parameters of the CSA which makes them dynamic in the dCSA. The dynamic step size diversifies the solution search for sufficient exploration, while the dynamic control probability guides the evolution of dCSA towards obtaining the global optimal value of the optimization benchmark functions by ensuring proper balance between exploration and exploitation.

Table I. Performance Evaluation of CSA over DCSA

S/N	Test Functions	Global Minimal	CSA	dCSA
1	Ackley	0.0000E+00	9.1363E-06	**8.8818E-16**
2	Dejong	0.0000E+00	3.2228E-06	**3.016E-260**
3	Easom	-1.0000E+00	-0.5338E+00	**-1.0000E+00**
4	Greiwangk	0.0000E+00	3.0067E-06	**0.0000E+00**
5	Michalwicz	-9.6602+00	-2.9949E+00	**-10.9913E+00**
6	Rastrigin	0.0000E+00	**9.6917E-06**	9.1745E+00
7	Rosenbrock	0.0000E+00	8.2910E-06	**0.0000E+00**
8	Shwefel	-4.1898E+02	-3.9020E+00	**-3.9337E+00**

9	Shubert	-1.8673E+02	-5.7704E+01	**-1.4505E+02**
10	Sphere	0.0000E+00	9.7799E-06	**1.1681E-21**

Note that algorithm with the best performances with respect to the global solution are shown in bold in the above table.

Table I can be understood more in the context of the ability of each of the algorithm to obtain an optimum result or close to an optimum result by 3D visualizations of the results. Figures 12 – 16 presents the 3D visualizations.

When Ackley and De Jong function where compared with respect to the performance of both algorithm in optimizing the functions, the results obtained by the algorithms were presented in Table I and plot of such result is shown in Figure 12.

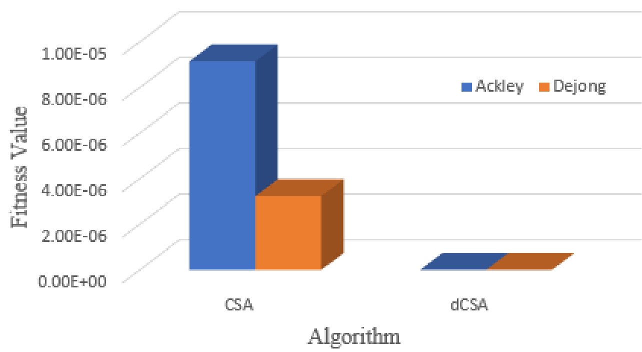

Figure 12. 3D Plot of Ackley and De Jong Function

From Figure 12, the more the fitness value move close to zero, the better the performance of the algorithm. Thus, it is obvious that dCSA outperforms CSA in optimizing these benchmark functions.

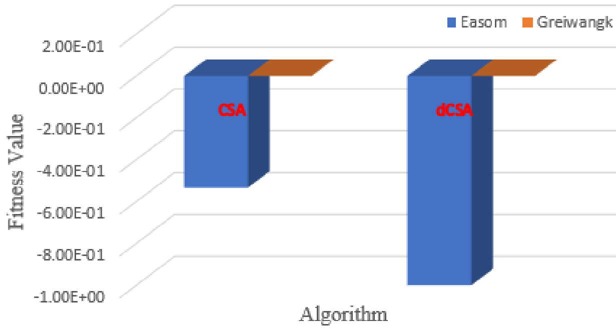

Figure 13. 3D Plot of Easom and Griewangk Function

Figure 13 present the plot of Easom and Griewangk function each with a global optimum of -1 and 0 respectively. It is obvious from the plot that dCSA outperforms CSA by obtaining the optimum results.

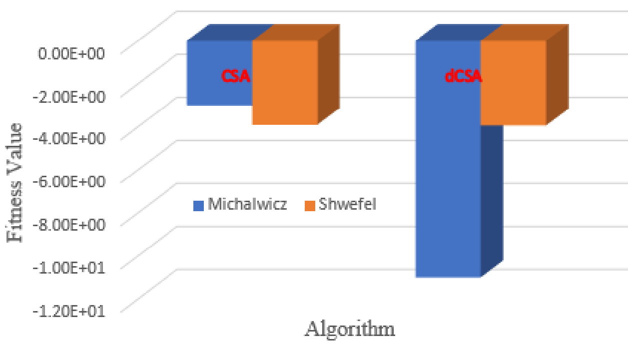

Figure 14. 3D Plot of Michalwicz and Rastrigin Function

Figure 14 present the plot of Michalwicz and Shwefel function each with a global optimum of -9.66 and respectively. For both benchmark functions, the more the algorithm obtain a higher negative value of fitness, the better the results. Thus, it is obvious from the plot that dCSA outperforms CSA for optimizing these functions obtaining a closed to optimum result.

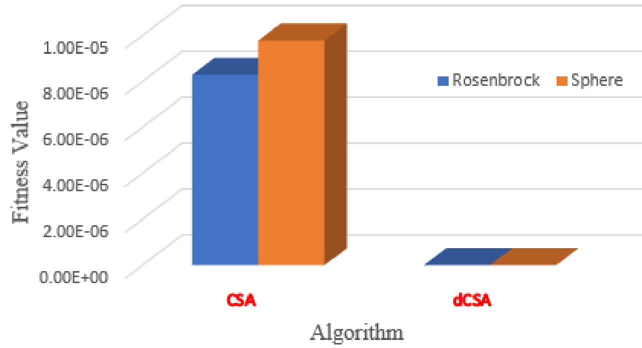

Figure 15. 3D Plot of Rosenbrock and Sphere Function

Figure 15 present the plot of Rosenbrock and Sphere function each with a global optimum of 0. It is obvious from the plot that dCSA outperforms CSA by obtaining the optimum result of these two benchmark functions.

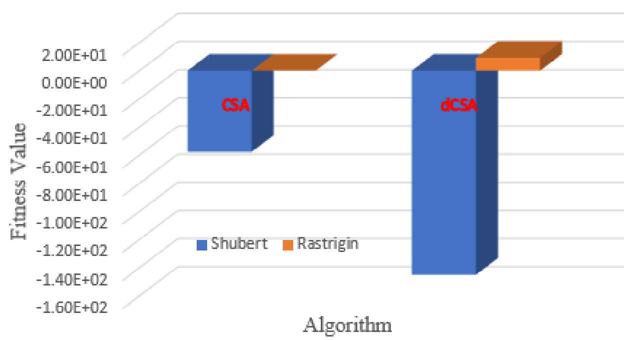

Figure 16. 3D Plot of Shubert and Rastrigin Function

Figure 16 present the plot of Shubert and Rastrigin function each with a global optimum of -1.86E+02 and 0 respectively. For Shubert function, the more the algorithm obtain a higher negative value of fitness, the better the result. Thus, it is obvious from the plot that dCSA outperforms CSA for optimizing Shubert function. However, CSA outperforms dCSA by optimizing Rastrigin function in obtaining a closed to the global optimum of 0.

5. Conclusion

The dynamic cuckoo search algorithm (dCSA) was developed and implemented in MATLAB R2013b. The performance of the developed algorithm was evaluated using ten benchmark optimization functions. These functions are categorized into unimodal and multimodal benchmark optimization functions (Ackley, De Jong, Easom, Rosenbrock, Griewangk, Michalwicz, Rastrigin Rosenbrock, Shwefel, Shubert and Sphere). The simulation results obtained shows that dCSA performed better when compared with the standard CSA in terms of precision accuracy and better quality results. This gives it more ability of escaping local minima than the standard CSA.

Acknowledgement

The authors wish to express their appreciation and gratitude to the Control and Computer Research Group of the Department of Computer Engineering, ABU, Zaria, Nigeria and the entire members in the Tuesday meetings, where academic discourse are at its best.

References

Bansal, J. C., Singh, P., Saraswat, M., Verma, A., Jadon, S. S., & Abraham, A. (2011). Inertia weight strategies in particle swarm optimization. Paper presented at the Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on.

Chauhan, P., Deep, K., & Pant, M. (2013). Novel inertia weight strategies for particle swarm optimization. Memetic computing, 5(3), 229-251.

Chu, S.-C., Tsai, P.-W., & Pan, J.-S. (2006). Cat swarm optimization. Paper presented at the Pacific Rim International Conference on Artificial Intelligence.

Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artificial intelligence review, 1-32.

Dorigo, M., & Thomas, S. (2004). Ant Colony Optimization. Cambridge, vol. 9, Dec. 2002: MIT Press.

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. Paper presented at the Proceedings of the sixth international symposium on micro machine and human science.

El Aziz, M. A., & Hassanien, A. E. (2016). Modified cuckoo search algorithm with rough sets for feature selection. Neural Computing and Applications, 1-10.

Fister Jr, I., Fister, D., & Fister, I. (2013). A comprehensive review of cuckoo search: variants and hybrids. International Journal of Mathematical Modelling and Numerical Optimisation, 4(4), 387-409.

Haruna, Z., Mu'azu, M. B., Abubilal, K. A., & Tijani, S. A. (2017). Development of a modified bat algorithm using elite opposition—Based learning. Paper presented at the Electro-Technology for National Development (NIGERCON), 2017 IEEE 3rd International Conference on.

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation, 4(2), 150-194.

Kamat, S., & Karegowda, A. (2014). A

brief survey on cuckoo search applications. Int. J. Innovative Res. Comput. Commun. Eng, 2(2).

Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K., & China, H. (2013). Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization. gene, 7(33), 8.

Li, X., Wang, J., & Yin, M. (2014). Enhancing the performance of cuckoo search algorithm using orthogonal learning method. Neural Computing and Applications, 24(6), 1233-1247.

Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. Test functions for optimization needs.

Ojha, R., & Das, M. (2012). An Adaptive Approach for Modifying Inertia Weight using Particle Swarm Optimisation. IJCSI International Journal of Computer Science Issues, 9(5), 105-112.

Qu, C., & He, W. (2015). A Double Mutation Cuckoo Search Algorithm for Solving Systems of Nonlinear Equations. International Journal of Hybrid Information Technology, 8(12), 433-448.

Shehab, M., Khader, A. T., & Al-Betar, M. A. (2017). A survey on applications and variants of the cuckoo search algorithm. Applied Soft Computing.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. Paper presented at the Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.

Tang, K., Yáo, X., Suganthan, P. N., MacNish, C., Chen, Y.-P., Chen, C.-M., & Yang, Z. (2007). Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Nature Inspired Computation and Applications Laboratory, USTC, China.

Vaijayanthi, P., Yang, X.-S., Natarajan, A., & Murugadoss, R. High Dimensional Data Clustering Using Cuckoo Search Optimization Algorithm. International Journal of Advanced Computer Engineering and Communication Technology, 3, 31-35.

Valian, E., Mohanna, S., & Tavakoli, S. (2011). Improved Cuckoo Search Algorithm for Global Optimization. International Journal of Communications and Information Technology, IJCIT, 1, 31-44.

Weisstein, E. W. (2002). Sigmoid function.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. Paper presented at the International Symposium on Stochastic Algorithms.

Yang, X.-S. (2010a). Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation, 2(2), 78-84.

Yang, X.-S. (2010b). A new metaheuristic bat-inspired algorithm Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74): Springer.

Yang, X.-S. (2012). Nature-inspired mateheuristic algorithms: Success and new challenges. arXiv preprint arXiv:1211.6658.

Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. Paper presented at the Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.

Yang, X.-S., & Deb, S. (2014). Cuckoo search: recent advances and applications. Neural Computing and Applications, 24(1), 169-174.

Yılmaz, S., & Küçüksille, E. U. (2015). A new modification approach on bat algorithm for solving optimization problems. Applied Soft Computing, 28, 259-275.

Zhang, Z., Sheng, Z., Shi, H., & Fan, Z. (2016). Inversion for refractivity parameters using a dynamic adaptive Cuckoo Search with crossover operator algorithm. Computational intelligence and neuroscience, 2016.