



An adaptive Cuckoo search algorithm for optimisation



M. Mareli^{a,*}, B. Twala^b

^a Department of Electrical and Electronics Engineering, University of Johannesburg, PO Box 542, Auckland Park 2006, South Africa

^b Institute of Intelligent Systems, Department of Electrical and Electronics Engineering University of Johannesburg, PO Box 542, Auckland Park 2006, South Africa

ARTICLE INFO

Article history:

Received 18 May 2017

Revised 30 August 2017

Accepted 3 September 2017

Available online 6 September 2017

Keywords:

Cuckoo search

Optimisation

Random walk

Lévy distribution

Pareto distribution

Test functions

ABSTRACT

Cuckoo search is one of many nature-inspired algorithms used extensively to solve optimisation problems in different fields of engineering. It is a very effective in solving global optimisation because it is able to maintain balance between local and global random walks using switching parameter. The switching parameter for the original Cuckoo search algorithm is fixed at 25% and not enough studies have been done to assess the impact of dynamic switching parameter on the performance of Cuckoo search algorithm. This paper's contribution is the development of three new Cuckoo search algorithms based on dynamically increasing switching parameters. The three new Cuckoo search algorithms are validated on ten mathematical test functions and their results compared to those of Cuckoo search algorithms with constant and dynamically decreasing switching parameters respectively. Finally, the simulations in this study indicate that, the Cuckoo search algorithm with exponentially increasing switching parameter outperformed the other Cuckoo search algorithms.

© 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Contents

1. Introduction	107
2. Cuckoo search	108
2.1. Cuckoo breeding behaviour	108
2.2. Lévy flights	108
2.3. Cuckoo search algorithm	108
2.4. Some Cuckoo search algorithm improvements	109
2.4.1. Step size improvements	109
2.4.2. Other improvements	109
3. Proposed Cuckoo search	109
4. Results and discussions	110
4.1. Simulation setup	110
4.2. Characteristics of test functions	114
4.3. Simulations results and discussions	114
5. Conclusions	114
Acknowledgements	114
References	114

* Corresponding author.

E-mail addresses: mmareli@hotmail.com (M. Mareli), btwala@uj.ac.za (B. Twala).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

1. Introduction

Optimisation plays an important role in solving different engineering problems. The goal of optimisation process is to determine

<http://dx.doi.org/10.1016/j.aci.2017.09.001>

2210-8327/© 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

either a maximum or a minimum value of the problem being solved, generally known as the objective function [1]. These problems include but not limited to systems design, electricity network operation, electricity generation, wireless communications routing and minimisation of energy losses during electricity transmission. Proper validations of optimisation algorithms require assessment of computational time and convergence rate in addition to the accuracy to determine the minimum or maximum values [2–8].

Some researchers have innovated optimisation algorithms based on nature observations, these algorithms are known as nature-inspired algorithms. In [9] a bat-inspired algorithm was developed based on echolocation to sense distance between a bat and its surroundings. Particle swarm optimisation (PSO) was innovated after observing behaviour of fish and birds schooling [10]. Differential evolution (DE) algorithm was created by Storm and Prince [11] based on population evolution using mutation, crossover and selection operations. Ant and bee algorithms were developed based on the foraging behaviour of ants and bees that use pheromone as a chemical messenger and the concentration of pheromone is regarded as indication of quality solutions to the problem being solved [12]. Simulated annealing (SA) is based on the characteristics of the metal annealing process [13].

Cuckoo search (CS) algorithm is also a nature-inspired algorithm, based on brood reproductive strategy of cuckoo birds to increase their population [14]. However, CS is more effective than other nature-inspired algorithms. In fact, DE, SA and PSO are special cases of CS algorithm, hence it is no surprising why CS algorithm outperforms them [14]. In [15] CS algorithm outperformed DE algorithm in terms of convergence speed to reach optimum solution. In addition, CS algorithm was reported as being more computationally efficient than the PSO [16].

Baskan [17] used CS algorithm to minimise traffic congestion by improving the performance of transportation road networks. The objective function was defined as a total of travel time and invested cost of 16 link capacity expansions. Cuckoo search produced best results when compared to other methods found in the literature. In [18] CS algorithm was used to maintain fault level and the voltage fluctuations within acceptable level, thus minimise real power losses in a smart grid.

The aim of this paper, is to develop and assess the performance of Cuckoo search algorithms with increasing switching parameter between local and global random walks as the number of iteration increases. The first CS algorithm has linearly increasing switching parameter. The second CS algorithm implements exponentially increasing switching parameter, and the third CS algorithm uses increasing power switching parameter.

The rest of this paper is organised as follows: Section 2 discusses the Cuckoo bird's breeding behaviour, Lévy flights and CS algorithm and some of CS algorithm improvements carried out by other scholars to make CS algorithm even more effective. Three new CS algorithms are proposed in Section 3. Simulation results and their interpretations are discussed in Section 4. Finally, the paper conclusions are presented in Section 5.

2. Cuckoo search

2.1. Cuckoo breeding behaviour

Cuckoos are a family of birds with unique reproductive strategy more aggressive compared to other bird's species. Some of cuckoo bird's species like Ani and Guira lay eggs in communal nests; however, they may remove others' eggs to increase the hatching probability of their own eggs. Other species use brood parasitism method of laying their eggs in the nests of other birds or host nests [19].

The parasitic cuckoos are good in sporting nests where eggs have just been laid and their timing of laying eggs is very precise. They lay one egg in the host nest which will normally hatch quicker than the other eggs. When this happens, the foreign cuckoo would remove the non-hatched eggs from the nest by pushing the eggs out of the nest. This behaviour is aimed at reducing the probability of the legitimate eggs from hatching. Furthermore, the foreign cuckoo chick can gain access to more food by mimicking the call of the host chicks. There are times when the host cuckoo discovers that one of the eggs is foreign. In that case the cuckoo either gets rid of the egg or abandon the nest altogether and moves to build a new nest somewhere else [19].

2.2. Lévy flights

Lévy flights are random walks whose directions are random and their step lengths are derived from the Lévy distribution. These Lévy flights are performed by animals and insects and it is characterised by series of straight flights followed by sudden 90° turns. Compared to normal random walks, Lévy flights are more efficient in exploring large-scale search areas. That is mainly due to Lévy flights variances increases much faster than that of the normal random walk. Lévy flights can reduce the number of optimisation algorithms iterations by about 4 orders compared to normal random walk [14].

2.3. Cuckoo search algorithm

Cuckoo search algorithm is a nature-inspired algorithm developed based on reproduction of cuckoo birds [14]. While working with CS algorithms, it is important to associate potential solutions with cuckoo eggs. Cuckoos normally lay their fertilised eggs in other cuckoos' nests with the hope of their off-springs being raised by proxy parents. There are times when the cuckoos discover that the eggs in their nests do not belong to them, in those cases the foreign eggs are either thrown out of the nests or the whole nests are abandoned. The CS optimisation algorithm is basically based on the following three rules:

- Each cuckoo selects a nest randomly and lays one egg in it.
- The best nests with high quality of eggs will be carried over to the next generation.
- For a fixed number of nests, a host cuckoo can discover a foreign egg with a probability $p_a \in [0,1]$. In this case, the host cuckoo can either throw the egg away or abandon the nest and build a new one somewhere else.

The last rule can be approximated by replacing a fraction p_a of the n host nests with new nests (with new random solutions). The quality or fitness of a solution can simply be proportional to the value of the objective function. From the implementation point of view, the representation that is followed is that each egg in a

Table 1
Local and Global random walk parameters.

Parameter	Description
x_i^t and x_k^t	Current positions selected by random permutation.
α	Positive step size scaling factor.
x_i^{t+1}	Next position.
s	Step size.
\otimes	Entry-wise product of two vectors.
H	Heavy-side function.
p_a	Used to switch between local and global random walks.
ε	Random number from uniform distribution.
$L(s, \lambda)$	Lévy distribution, used to define the step size of random walk

```

Objective function  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)^T$ 
Generate initial population of  $n$  host nests  $x_i$  ( $i = 1, 2, \dots, n$ )
While ( $t < \text{Max Generation}$ ) or (stop criteria)
  Get a cuckoo (say  $i$ ) randomly by Lévy distribution;
  Evaluate its quality/fitness  $F_i$ ;
  Choose a nest among  $n$  (say  $j$ ) randomly;
  Evaluate its quality/fitness  $F_j$ ;
  If ( $F_i > F_j$ )
    Replace  $j$  by the new solution;
  End
  A fraction of ( $P_a$ ) of worse nests are abandoned and
  new ones are built at new locations via Lévy flights;
  Keep the best solutions (or nests with quality solutions);
  Rank the solutions and find the current best;
End while
Post processing

```

Fig. 1. Lévy based Cuckoo search pseudo code for a global optimisation.

nest represents a solution, and each cuckoo can lay only one egg (thus representing one solution). We can safely make no deference between an egg, a nest or a cuckoo. The aim is to use the new and potentially better solution (cuckoo egg) to replace a bad solution in the nest.

Cuckoo search algorithm is very effective for global optimisation problems since it maintains a balance between local random walk and the global random walk. The balance between local and global random walks is controlled by a switching parameter $p_a \in [0, 1]$. The local and global random walks are defined by Eqs. (1) and (2), respectively. Their parameters are defined in Table 1.

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

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

The basic steps of the Cuckoo search algorithm based on three rules can be summarised as pseudo code shown in Fig. 1.

2.4. Some Cuckoo search algorithm improvements

2.4.1. Step size improvements

This section discusses research studies done to improve the efficiency of Cuckoo search, these studies apply different probability distributions to determine the Cuckoo search random walk step sizes.

The first study was done by Zheng and Zhou [20], they used Gauss distribution instead of original Lévy distribution to determine the random walk step size. When applied to find global minimum values of 6 mathematics test functions, the Gauss CS performed better than the Lévy CS for all cases. Furthermore, the Gauss and Lévy CS algorithms were used to solve engineering design optimisation problem. The results further confirmed that Gauss CS is better than the Lévy CS in terms of higher convergence rate and the average generation was reduced from 20.15 to 13.95 for Gauss CS.

The rapid growing rate of documentation in the Internet space posed some challenges especially in the documentation retrievals

process. Zaw and Mon [21], solved this web document clustering by using a Gauss based CS algorithm. The algorithm was tested on 3 clusters and 300 documents. The results confirmed that Gauss CS algorithm outperformed Lévy CS algorithm. More specifically, the convergence rate of Gauss CS and Lévy CS are 120 and 160 iterations, respectively. The quality of clustering was determined by a combination of Precision and Recall, called F-measure where high F-measure indicated high accuracy. The Gauss CS algorithm and Lévy CS algorithm produced F-measure of 0.626 and 0.619, respectively.

Ho et al. [22] proposed CS algorithm using Gaussian and Cauchy distributions and applied them to solve economic emission load dispatch problem with multiple fuel options. The new versions of CS algorithms resulted in fewer parameters, fewer equations and shorter computational processes when compared to Lévy CS. In addition, the Gauss CS performed better than the Cauchy CS algorithm. The application of Gauss CS and Cauchy CS to short term hydrothermal scheduling with reservoir volume constraint was done by Nguyen et al. [23]. In this study, however, Lévy CS produced the best results with lowest minimum compared to Gauss and Cauchy CS algorithms. Furthermore, the Gauss CS algorithm average time was 1.47% more than the Lévy CS algorithm average time. While the Cauchy CS algorithm average time was 4.83% more than that of Lévy CS algorithm.

Roy et al. managed to improve CS by using Gamma distribution instead of original Lévy distribution. When tested on 6 mathematical test functions, the Gamma based CS proved to be more accurate and efficient than the Lévy CS algorithm [24]. The best performance was recorded for the Ackley test function for 1000 iterations where Lévy and Gamma CS algorithms produced average minimum values of $1.0923 \exp(-15)$ and $2.22507 \exp(-308)$, respectively.

2.4.2. Other improvements

In the original cuckoo search algorithm, switching parameter between local and global random walks is keep constant at 0.25 as Cuckoo search algorithm searches for global minima. There is limited research studies in CS algorithm efficiency improvement by dynamically changing the value of switching parameter. In [25] the value of switching parameter was linearly decreased as CS algorithm was searching for global minimum value of the test function. Another CS improvement was achieved by selecting suitable nest by using sorting function instead on permutation function [26].

3. Proposed Cuckoo search

In the previous Section 2, CS efficiency improvements were accomplished by determining the CS random step size using other probability distribution functions other than the original Lévy distribution. In this section three CS algorithms are proposed based on dynamic increasing switching parameter as the number of CS iterations increases. The new CS algorithms are defined as per Eqs. (3)–(5) and their parameters are summarised in Table 2.

Table 2
Dynamic Switching parameter definition.

Parameter	Description
p_{aCi}	Switching parameter for the current iteration.
p_{aMax}	Maximum value of switching parameter.
C_i	Current iteration.
T_i	Set total number of iteration.
Exp	Mathematical exponential function

Table 3

Different Cuckoo search algorithms based on changing switching parameter.

Cuckoo search	Description
CSCo	This Cuckoo search uses constant switching parameter.
CSLD	Cuckoo search using linear decreasing switching parameter.
CSLI	Cuckoo search using linear increasing switching parameter.
CSEI	Cuckoo search using exponential increasing switching parameter.
CSPI	Cuckoo search using power increasing switching parameter.

Table 4

Test functions modality and global minimum values.

Function	Modality	Global minimum value	References
Ackley	Multimodal	0	[29]
Griewank	Multimodal	0	[27]
Bohachevsky	Multimodal	0	[14]
De Jong	Unimodal	0	[30]
Matyas	Unimodal	0	[14]
Zakharov	Unimodal	0	[14]
Goldstein-Prices	Multimodal	3	[14]
Rosenbrock	Unimodal	0	[31]
Easom	Unimodal	−1	[14]
Michalewicz	Multimodal	−9.66	[32]

The first proposed CS algorithm uses switching parameter whose value is linearly increased as the number of CS iterations increases. The switching parameter is defined by Eq. (3).

$$p_{aCi} = (p_{aMax}) * (C_i/T_i) \quad (3)$$

Eq. (4) corresponds to switching parameter that is exponentially increased as iterations increases.

$$p_{aCi} = (p_{aMax}) * \text{Exp}(C_i/T_i) \quad (4)$$

Eq. (5) represents switching parameter that is increased in a power of three as iterations increases.

$$p_{aCi} = (p_{aMax}) * (C_i/T_i)^3 \quad (5)$$

4. Results and discussions

4.1. Simulation setup

Five Cuckoo search algorithms tabulated in Table 3 were investigated in MATLAB version 7.10.0.499 (R201a). The test environment was FUJITSU laptop with the following specifications; RAM of 3.0 GB, CPU is Intel Celeron 900@2.2 Ghz and 32 bit windows 7 home Basic operating system.

The number of nests, n was set to 25 for all CS algorithms since any number between 15 and 40 is sufficient for most optimisation

Table 5

CSCo results.

Test Function	Global minimum value
Ackley	1.10E-07
Griewank	1.47E-06
Bohachevsky	0
De Jong	1.31E-17
Matyas	6.60E-42
Zakharov	9.79E-10
Goldstein-Prices	3
Rosenbrock	8.12E-01
Easom	−1
Michalewicz	−7.82

Table 6

CSLD results.

Test Function	Global minimum value
Ackley	9.67E-08
Griewank	1.97E-06
Bohachevsky	0
De Jong	6.44E-18
Matyas	6.75E-46
Zakharov	2.90E-12
Goldstein-Prices	3
Rosenbrock	1.31E-01
Easom	−1
Michalewicz	−7.39

Table 7

CSLI results.

Test Function	Minimum value
Ackley	3.35E-08
Griewank	3.72E-08
Bohachevsky	0
De Jong	7.26E-18
Matyas	1.22E-46
Zakharov	1.17–13
Goldstein-Prices	3
Rosenbrock	1.97E-01
Easom	−1
Michalewicz	−7.01

Table 8

CSEI results.

Test Function	Minimum value
Ackley	5.94E-07
Griewank	2.83E-07
Bohachevsky	0
De Jong	5.83E-17
Matyas	1.04E-33
Zakharov	1.69E-07
Goldstein-Prices	3
Rosenbrock	2.19
Easom	−1
Michalewicz	−8.87

Table 9

CSPI results.

Test Function	Minimum value
Ackley	2.55E-08
Griewank	1.13E-07
Bohachevsky	0
De Jong	5.90E-18
Matyas	6.75E-50
Zakharov	4.00E-16
Goldstein-Prices	3
Rosenbrock	5.58E-01
Easom	−1
Michalewicz	−6.93

Table 10

95% Confidence Interval (CI).

CS	CI	CI Length
CSCo	−2.48495 < μ < 1.48495	3.9699
CSLD	−2.401378 < μ < 1.349378	3.750756
CSLI	−2.279265 < μ < 1.316665	3.59593
CSEI	−2.695006 < μ < 1.81006	4.505066
CSPI	−2.2280448 < μ < 1.3540448	3.5820896

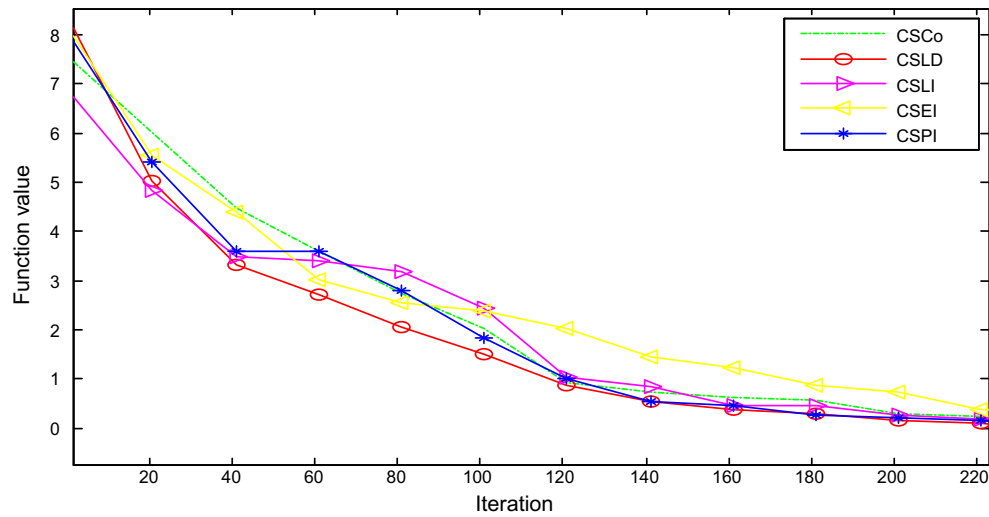


Fig. 2. Cuckoo search algorithms average convergences for Ackley function.

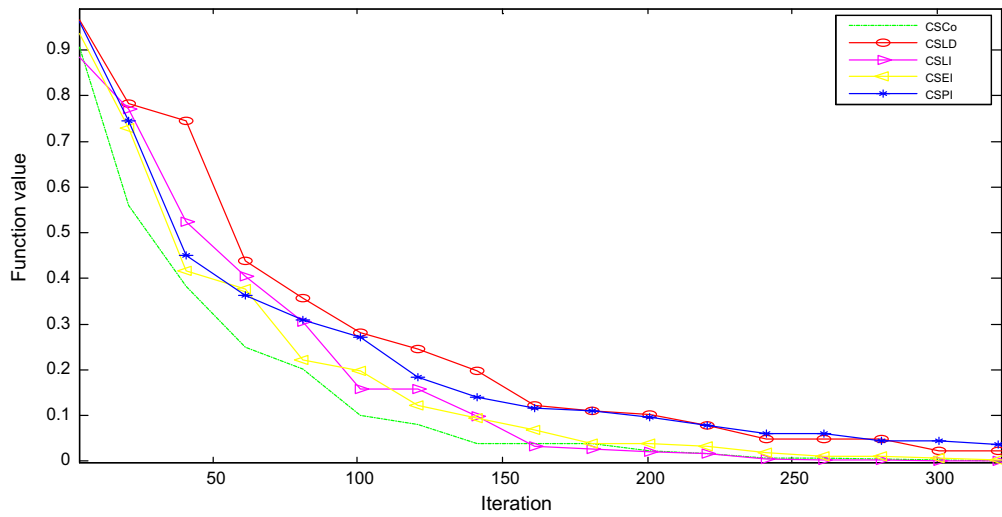


Fig. 3. Cuckoo search algorithms average convergences for Griewank function.

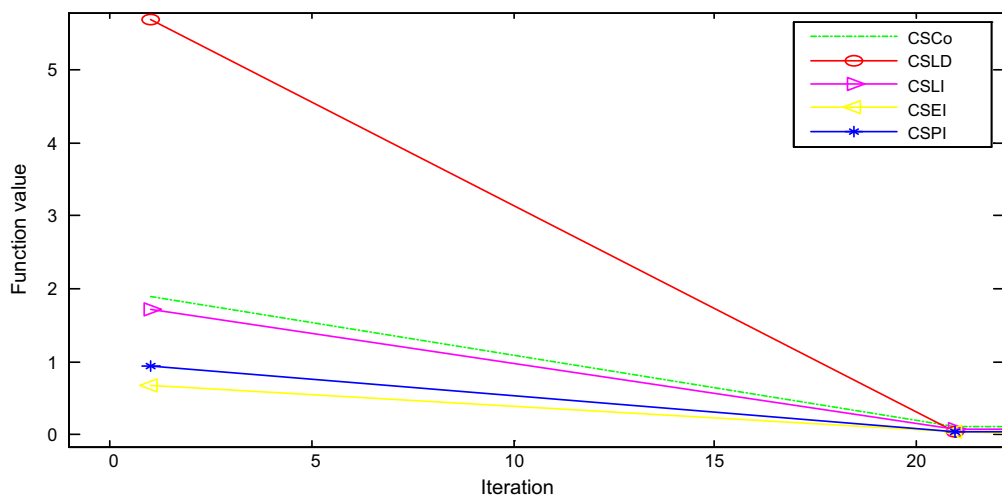


Fig. 4. Cuckoo search algorithms average convergences for Bohachevsky function.

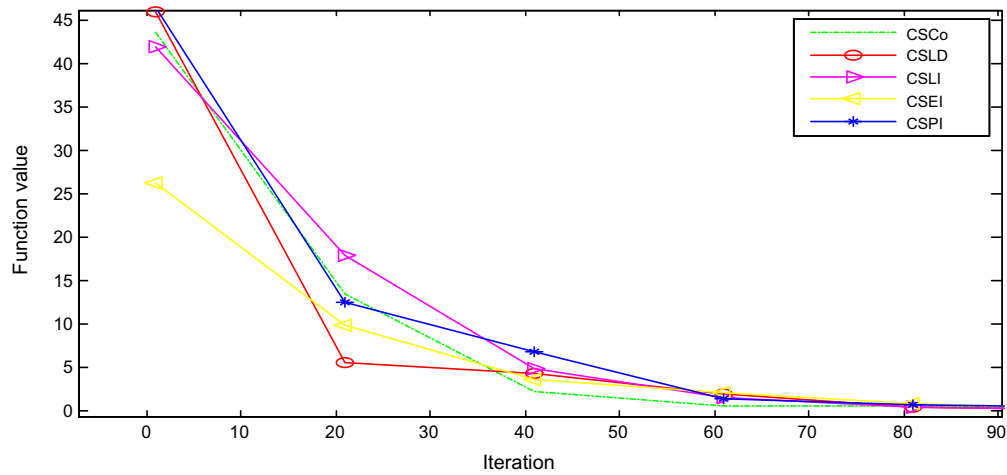


Fig. 5. Cuckoo search algorithms average convergences for De Jong function.

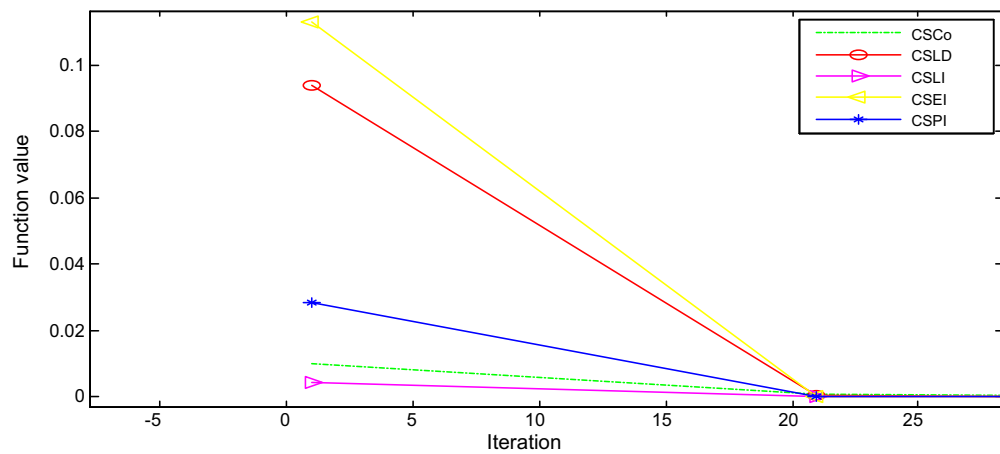


Fig. 6. Cuckoo search algorithms average convergences for Matyas function.

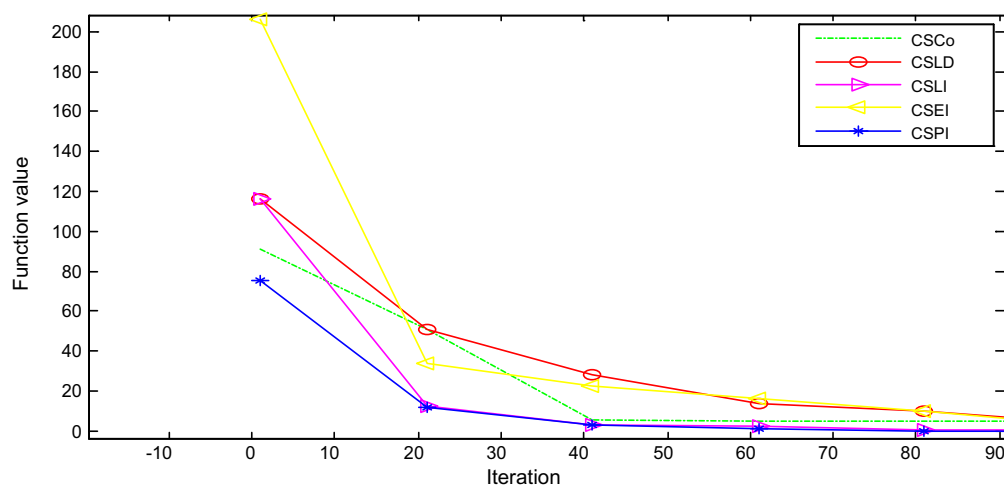


Fig. 7. Cuckoo search algorithms average convergences for Zakharov function.

problems [27]. The step size scaling factor was set to 0.01 while the total of iterations was 1000 and the value of p_{aMax} was 0.25.

Due to the stochastic nature of CS algorithms, the start points for the algorithms are not the same and the paths followed are also

different. To address these differences, each CS algorithm was run twenty times for each of the ten test functions and the results were saved in Microsoft excel file. Using Microsoft Excel, statistical analysis was performed to determine mean, standard deviation. Then

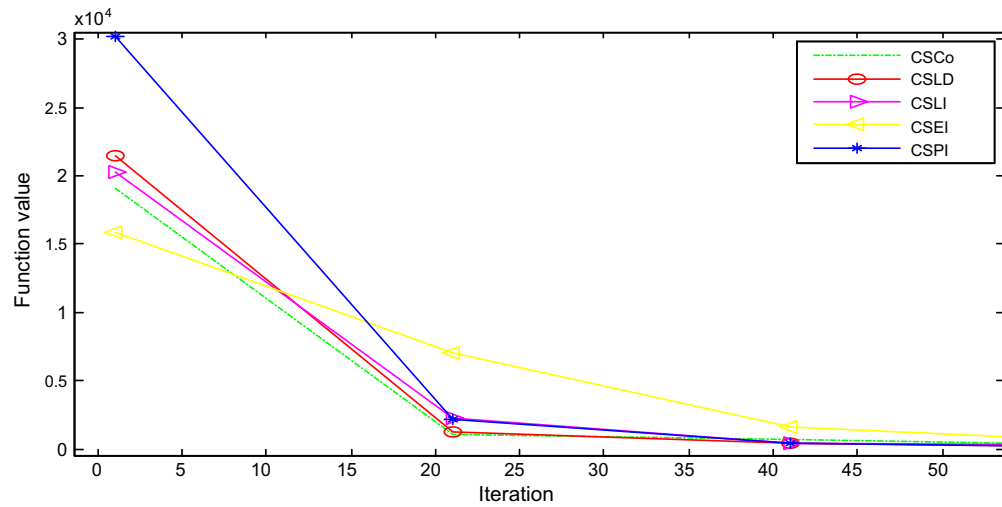


Fig. 8. Cuckoo search algorithms average convergences for Rosenbrock function.

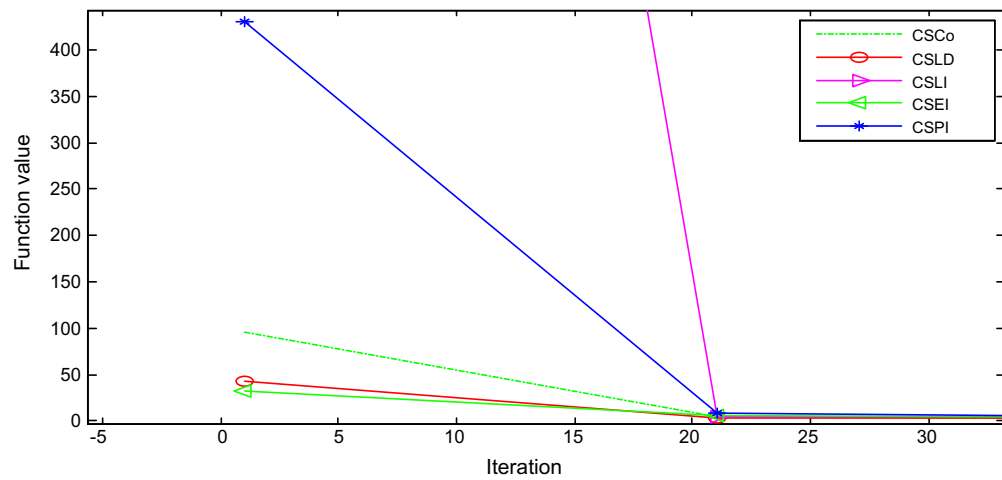


Fig. 9. Cuckoo search algorithms average convergences for Goldstein-Prices function.

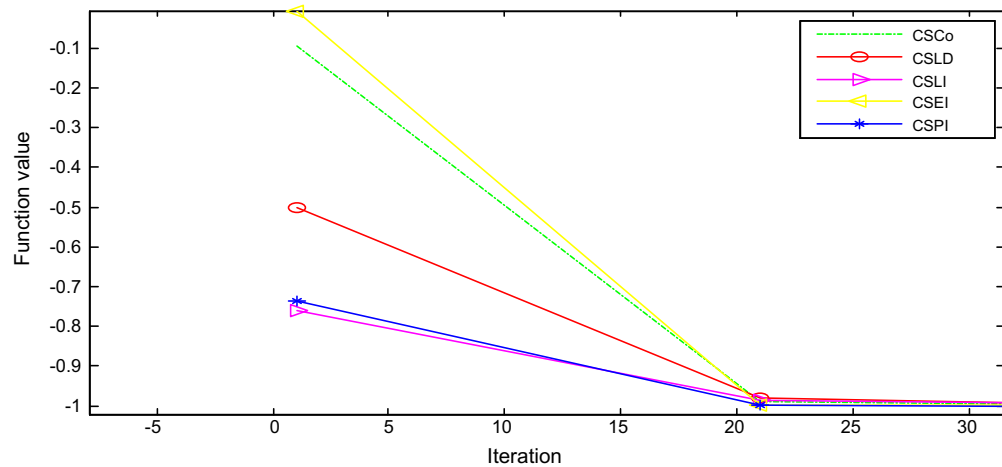


Fig. 10. Cuckoo search algorithms average convergences for Easom function.

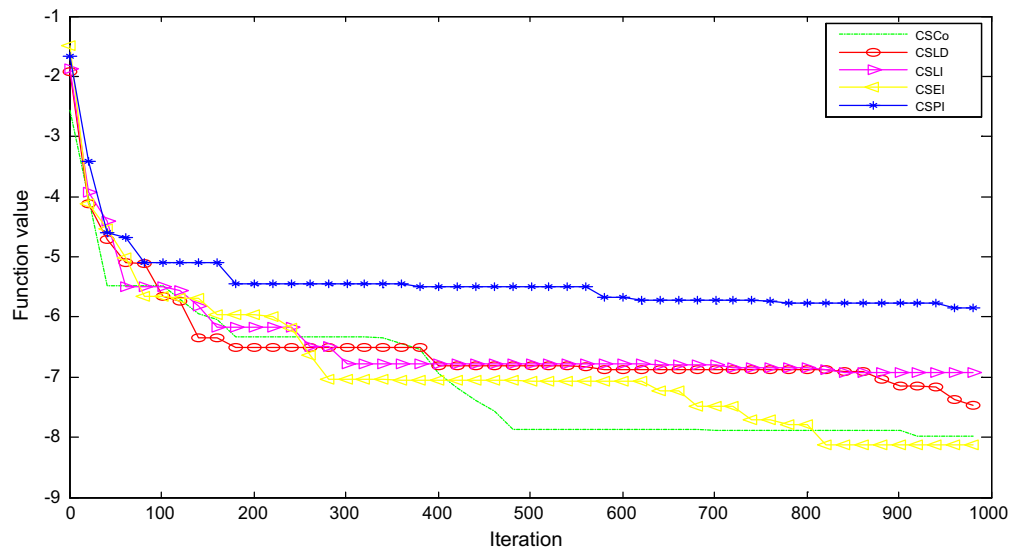


Fig. 11. Cuckoo search algorithms average convergences for Michalewicz function.

Table 11
CS convergence.

CS Version	Plot	Number of leading
CSLI	3, 6 and 10	3
CSEI	4, 9 and 11	3
CScO	5 and 8	2
CSPI	7	1
CSLD	2	1

95% confidence interval (CI) for each CS algorithm was determined and the CS algorithm with larger CI is deemed as more efficient than the one with smaller CI interval.

4.2. Characteristics of test functions

The benchmark test functions used to validate the performance of each CS are tabulated in Table 4. Test functions modality is defined as the number of peaks encountered in the function landscape. These peaks can negatively impact the optimisation process when the optimisation algorithm gets trapped in the peaks [28]. Some test functions are unimodal while others are multimodal.

4.3. Simulations results and discussions

The obtained global minimum values for each of CS algorithm tested are tabulated in Tables 5–9. The corresponding 95% confidence intervals were calculated and tabulated in Table 10.

From Table 10, the CSEI has the longest confidence interval (4.505066) as per CI Length column. That means that CSEI is more effective in finding global minimum values of the sample to test functions used. The second most effective algorithm is CScO with confidence interval of 3.9699, the original CS with constant switching parameter. The least effective algorithm is CSLI with lowest confidence interval of 3.59593.

Cuckoo search convergence results depicted in Figs. 2–11. Some iteration axis have been resized to reveal details.

The convergence results for Figs. 2–11 are summarised in Table 10. This is more of an indication as to how different CS algorithms converge to the global minimal point for each test function used (see Table 11).

5. Conclusions

This paper introduced nature-inspired optimisation algorithms and highlighted that CS algorithm is a more general and efficient algorithm when compared to PSO, DE and SA. Then some studies carried out to improve the CS efficiency were reviewed. It was found that most CS efficiency improvements were based on applying different probability distribution functions (Cauchy, Gauss and Gamma) to determine the random walk step sizes. The contribution of this paper is the investigation of dynamically increasing switching parameter in CS algorithms performance. Three CS algorithms using linear, exponential and power increasing switching parameters were developed and tested against the constant and linearly decreasing CS algorithms. The CS using exponential increasing (CSEI) switching parameter was found to be more efficient than other CS algorithms.

Acknowledgements

This work is supported by the Institute for Intelligent Systems at the University of Johannesburg in South Africa. We are very thankful of Prof Xin-She Yang for providing demo CS algorithm in the book *Nature-Inspired Optimisation Algorithms*. We would like to thank also anonymous reviewers for their valuable comments.

References

- [1] S. Noureddine, An optimization approach for the satisfiability problem, *Appl. Comput. Inform.* 11 (1) (2015) 47–59.
- [2] E. Belic, N. Lukac, K. Dezelak, B. Zalik, G. Stumberger, GPU-based online optimization of low voltage distribution network operation, *IEEE Trans. Smart Grid* 8 (3) (2017) 1460–1468.
- [3] A. Koppel, B.M. Sadler, A. Ribeiro, Proximity without consensus in online multiagent optimization, *IEEE Trans. Signal Process.* 65 (12) (2017) 3062–3077.
- [4] A. Platonov, Information theory and optimization of analog feedback communication systems, in: 2016 IEEE International Black Sea Conference on Communications and Networking, 2016.
- [5] M. Tang, L. Gao, H. Pang, J. Huang, L. Sun, Optimizations and economics of crowdsourced mobile streaming, *Fog Comput. Netw.* (2017) 21–27.
- [6] S. Das, J.R. Doppa, P.P. Pande, K. Chakrabarty, Design-space exploration and optimization of an energy-efficient and reliable 3-D small-world network-on-chip, *IEEE Trans. Comp.-Aided Des. Integr. Circ. Syst.* 36 (5) (2017) 719–732.
- [7] Z. Yan, J. Fan, J. Wang, A collective neurodynamic approach to constrained global optimization, *IEEE Trans. Neural Learn. Syst.* 28 (5) (2017) 1206–1215.

- [8] A.R. Parkinson, R.J. Balling, J.D. Hedengren, *Optimization Methods for Engineering Design: Applications and Theory*, fifth ed., Brigham Young University, 2013.
- [9] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: J.R. Gonzalez, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, Springer, Berlin, 2010, pp. 65–74.
- [10] D.P. Rini, S.M. Shamsuddin, S.S. Yuhaniz, Particle swarm optimization: technique, system and challenges, *Int. J. Comput. Appl.* 14 (1) (2011) 19–27.
- [11] R. Storm, K. Prince, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341–359.
- [12] M. Kefayat, A.L. Ara, S.N. Niaki, A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources, *Energy Convers. Manage.* 92 (2015) 149–161.
- [13] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (4598) (1983) 670–680.
- [14] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, first ed., Elsevier, London, 2014.
- [15] M.I. Solihin, M.F. Zani, Performance comparison of Cuckoo search and differential evolution algorithm for constrained optimization, in: *International Engineering Research and Innovation Symposium (IRIS)*, vol. 160(1), 2016, pp. 1–7.
- [16] M.A. Adnan, M.A. Razzaque, A comparative study of particle swarm optimization and Cuckoo search techniques through problem – specific distance function, in: *2013 International Conference on Information and Communication Technology (ICICT)*, Bandung, Indonesia, 2013.
- [17] O. Baskan, Determining optimal link capacity expansions in road networks using Cuckoo search algorithm with Levy flights, *J. Appl. Math.* 2013 (2013) 1–11.
- [18] W. Buaklee, K. Hongesombut, Optimal DG allocation in a smart distribution grid using Cuckoo search algorithm, *ECTI Trans. Electr. Eng. Electron. Commun.* 11 (2) (2013) 16–22.
- [19] X.S. Yang, S. Deb, Engineering optimization by Cuckoo search, *Int. J. Math. Modell. Numer. Opt.* 1 (4) (2010) 330–343.
- [20] H. Zheng, Y. Zhou, A novel cuckoo search algorithm based on Gauss distribution, *J. Comput. Inf. Syst.* 8 (10) (2012) 4193–4200.
- [21] M.M. Zaw, E.E. Mon, Web document clustering using Gauss distribution based cuckoo search clustering algorithm, *Int. J. Sci. Eng. Technol. Res.* 3 (13) (2014) 2945–2949.
- [22] S.D. Ho, V.S. Vo, T.M. Le, T.T. Nguyen, Economic emission load dispatch with multiple fuel options using cuckoo search algorithm with Gaussian and Cauchy distributions, *Int. J. Energy, Inf. Commun.* 5 (5) (2014) 39–54.
- [23] T.T. Nguyen, D.N. Vo, B.H. Dinh, Cuckoo search algorithm using different distributions for short term hydrothermal scheduling with reservoir volume constraint, *Int. J. Electr. Eng. Inf.* 8 (1) (2016) 76–92.
- [24] S. Roy, A. Mallick, S.S. Chowdhury, S. Roy, A novel approach on cuckoo search algorithm using Gamma distribution, in: *Second International Conference on Electronics and Communication Systems*, 2015.
- [25] S.I. Tusi, N. Shawkat, M.A. Ahmed, B. Panday, N. Sakib, Comparative analysis on improved Cuckoo search algorithm and artificial Bee colony algorithm on continuous optimization problems, *Int. J. Adv. Res. Artif. Intell.* 4 (2) (2015) 14–19.
- [26] M. Tuba, M. Subotic, N. Stanarevic, Modified Cuckoo search algorithm for unconstrained optimization problems, in: *Proceedings of the European Computing Conference*, 2011.
- [27] X.-S. Yang, *Test Problems in Optimization*, in *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, 2010.
- [28] M. Jamil, X.S. Yang, A literature survey of benchmark functions for global optimization problems, *Int. J. Math. Model. Numer. Opt.* 4 (2) (2013) 150–194.
- [29] J.M. Dieterich, B. Hartke, Empirical review of standard benchmark functions using evolutionary global optimization, *Appl. Math.* 3 (10) (2012) 1552–1564.
- [30] N.A. Al-Madi, A.T. Khader, De Jong's sphere model test for a Social-Based Genetic Algorithm (SBGA), *Int. J. Comp. Sci. Netw. Security (IJCSNS)* 8 (3) (2008) 179–185.
- [31] N. Chase, M. Rademacher, E. Goodman, R. Averill and R. Siodhu, 2016. [Online]. Available: <http://www.yumpu.com/en/document/view/46603781/a-benchmark-study-of-optimization-search-algorithms-red-cedar-7/>. (Accessed 04 November 2016).
- [32] M. Molga, C. Smutnicki, "Robert Marks," Robert Marks, 2005. [Online]. Available: <http://www.robertmarks.org/Classes/papers>. (Accessed 29 April 2017).