Research Article

International Journal of Advanced Computer Research, Vol 7(32)
ISSN (Print): 2249-7277   ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2017.732006

Global optimisation using Pareto cuckoo search algorithm

Mahlaku Mareli[1*] and Bheki Twala[2]
Department of Electrical and Electronic Engineering, University of Johannesburg, South Africa[1]
Institute of Intelligent Systems, Department of Electrical and Electronic Engineering, University of Johannesburg, South Africa[2]

Received: 11-May-2017; Revised: 21-July-2017; Accepted: 25-July-2017

Abstract
Cuckoo search is one of nature-inspired algorithms successfully used for solving different optimisation problems. Cuckoo search has proved to be very effective than other nature-inspired algorithms, however, there is still room to improve it further by either by step sizes or probability of finding foreign egg. In this paper, we present an improved cuckoo search algorithm using a Pareto distribution instead of Levy distribution as per original cuckoo search. Five cuckoo search algorithms based on different distribution functions are developed and compared for performances, computational time and convergence rates. These cuckoo search algorithms performances were validated against ten standard test functions. It was found that the cuckoo search algorithm based on Pareto distribution outperformed other cuckoo search algorithms, i.e. Levy-based cuckoo search, Cauchy-based cuckoo search, Gauss-based cuckoo search and Gamma-based cuckoo search.

Keywords
Cuckoo search, Levy distribution, Cauchy distribution, Gamma distribution, Pareto distribution, Gauss distribution and test functions.

1.Introduction
It is important to reach an optimal way of doing things in the real world because resources are normally limited. Optimisation is about reaching better results using resources available. Some examples of optimisation include, but not limited to electricity network operation, electricity generation, wireless communications routing and minimization 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 [1-7].

Cuckoo search (CS) algorithms have proved to be more effective than other nature-inspired algorithms in solving complex problems [8, 9].

The original CS algorithm step sizes are derived from the Levy probability distribution function. However, some researchers have managed to improve the performance of CS by using different probability distributions to determine step sizes.

*Author for correspondence

The first study (in 2012) was done by Zheng and Zhou [10] who used Gauss distribution instead of Levy distribution. When applied to find global minimum values of 6 mathematics test functions, the Gauss CS performed better than the Levy CS in all cases. Furthermore, the Gauss and Levy CS algorithms were used to solve engineering design optimisation problem. The results further confirmed that Gauss CS is better than the Levy CS in terms of the 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 poses some challenges, especially in the documentation retrieval process. Zaw and Mon [11], 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 the Gauss CS algorithm outperformed Levy CS algorithm. More specifically, the convergence rate of Gauss CS and Levy CS are 120 and 160 iterations, respectively. The quality of clustering was determined by a combination of precision and recall, called the F - measure where high F-measure indicated high accuracy. The Gauss

CS algorithm and Levy CS algorithm produced F-measure of 0.626 and 0.619, respectively.

In 2014, Ho et al. [12] proposed CS by 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 Levy 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. [13]. In this study, however, Levy 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 Levy CS algorithm average time. While the Cauchy CS algorithm average time was 4.83% more than that of a Levy CS algorithm.

Roy et al. [14] managed to improve CS by using a Gamma distribution instead of original Levy distribution. When tested on 6 mathematical test functions, the Gamma based CS proved to be more accurate and efficient than the Levy CS algorithm. The best performance was recorded for the Ackley test function for 1000 iterations where Levy and Gamma CS algorithms produced average minimum valves of 1.0923exp (-15) and 2.22507exp (-308), respectively.

There are many probability distribution functions and the studies presented in the previous paragraphs, only used few probability functions to test their impact on the performance of the cuckoo search algorithm. The aim of this paper is to develop a Pareto based CS optimisation algorithm and compare its performance to current versions of CS algorithms.

The rest of this paper is organised as follows: section 2 introduces the cuckoo breeding behaviour, levy distribution and CS algorithm. Section 3 discusses the methodology adopted in comparing the performance of the CS algorithms, the benchmarking measures used and the test functions used in the experiments. The experiments and discussions are presented in section 4. Finally, section 5 concludes the paper and provides some of the further research suggestions regarding other probability distributions impact on CS performance.

2.Cuckoo search algorithm

2.1Cuckoo breeding behaviour

Cuckoos are a family of birds with a unique reproductive strategy more aggressive compared to other bird species. Some of cuckoo bird 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 [15].

Their timing of laying the eggs is very precise; the parasitic cuckoos are good in sporting nests where eggs have just been laid. 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 unhatched 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 [15].

2.2Levy distribution and Levy flights

Levy distribution is defined mathematically by (1),

$$L(s,\gamma,\mu) = \sqrt{\frac{\gamma}{2\pi}}\exp\left[-\frac{\gamma}{2(s-\mu)}\right]\frac{1}{(s-\mu)^{\frac{3}{2}}}\ , \qquad (1)$$

Where $0< \mu< s< \infty$ and γ is a scale parameter.

From (1), we can obtain generalised Levy distribution, (2) by increasing s towards infinity.

$$L(s,\gamma,\mu) \approx \sqrt{\frac{\gamma}{2\pi}}\frac{1}{(s)^{\frac{3}{2}}} \qquad (2)$$

Levy flights are random walks whose directions are random and their step lengths are derived from the Levy distribution. These Levy flights are performed by animals and insects and they are characterised by series of straight flights followed by sudden 90^{o} turns. Compared to normal random walks, Levy flights are more efficient in exploring large-scale search areas. That is mainly due to Levy flights variances increases much faster than that of the normal random walk.

Levy flights can reduce the number of optimisation algorithms iterations by about 4 orders compared to normal random walk [9].

2.3Cuckoo search algorithm

CS algorithm is a nature-inspired algorithm developed by Yang and Deb in 2009 [9] based on the reproduction on cuckoos birds. While working with CS algorithms, it is important to associate potential solutions with cuckoo eggs. Cuckoos normally lay their fertilized eggs in other cuckoos' nests with the hope of their offspring 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 $p_a \epsilon$ [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 assumption 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 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.

CS algorithm is very effective for global optimisation problems since it maintains a balance between local random walk and the global random walk. The switching between local and global random walks is controlled by a switching parameter $p_a \epsilon$ [0,1]. The local random walk is determined by (3).

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

Where;
x_i^{t+1} is next random walk position and x_i^t is current position;
x_j^t and x_k^t are different solutions selected by random permutation;
α is positive step size scaling factor, relates to the scales of the problem to be solved;
s is the step size;
$\otimes$ is the entry-wise product of two vectors;
H is a heavy-side function;
ε is random number from uniform distribution.

The global random walk is determined using Levy distribution by (4).

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \qquad (4)$$

Where;
The $\alpha > 0$ is the step size scaling factor, which is related to scales of the problem being solved.

$$L(s, \lambda) = \frac{\lambda\, \Gamma(\lambda) * \mathrm{Sin}\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{(1+\lambda)}}, \quad (s > 0) \qquad (5)$$

$\Gamma(\lambda)$ is known as a gamma function and defined in (6) [16].

$$\Gamma(\lambda) = \int_{-\infty}^{\infty} t^{\lambda-1} e^{-t}\, dt \qquad (6)$$

The integral in (6) converges for $\lambda \geq 1$
The gamma function can also be expressed in terms on factorial as $\Gamma(\lambda) = (\lambda-1)!$.

The basic steps of the CS algorithm based on three rules can be summarized as pseudo code shown in *Figure 1.*

```
Objective function $(fx)$,  $x = (x_1, x_2 \ldots\ldots x_d)^T$
Generate initial population of n host nests $xi$ ($i = 1,2..n$)
**While** (t < Max Generation) or (stop criteria)
   Get a cuckoo (say $i$ ) randomly by Pareto 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 ($P_a$ ) of worse nests are abandoned and
   new ones are built at new locations via Levy flights;
   Keep the best solutions (or nests with quality solutions);
   Rank the solutions and find the current best;
**End while**
Post processing
```

Figure 1 Levy based CS pseudo code for a global optimisation

3.Proposed Pareto based cuckoo search algorithm

We are proposing the use of Pareto distribution function in determining the global random walk for CS algorithm.

Pareto distribution is a skewed heavy tailed distribution named after Vilfredo Pareto. It was originally used to model the distribution of income in a society [17]. Pareto probability function is defined in (7).

$$f(x) = 1 - \frac{1}{x^{a}} \quad , \; x \in [1, \infty) \qquad (7)$$

Where a is called shape parameter and $1 \leq x$.
This function is increasing and continuous on [1, ∞], and it evaluates to $f(1) = 0$ and $f(x) \to 1$ as $x \to \infty$.

When $a = 1$, then Pareto distribution function is known as standard Pareto distribution.

Pareto cumulative distribution function is defined in (8).

$$f(x) = \frac{1}{x^{2}} \quad , \; x \in [1, \infty) \qquad (8)$$

The step size of Pareto random walk is represented as a Pareto quantile function defined in (9). This quantile function is an inverse function of cumulative distribution function [18].

$$F^{-1}(p) = \frac{1}{(1-p)^{1/2}} \; , \; p \in [0,1) \qquad (9)$$

For Pareto distribution, we determine the global random walk as;

$$x_i^{t+1} = x_i^t + \alpha * rand_3 * x_i^{new} \qquad (10)$$

Where $\alpha > 0$ is the step size multiplier and;

$$\mathrm{x_i^{new}} = \frac{1}{\sqrt{(1-\mathrm{rand_4})}} * (\mathrm{x_i^t} - \mathrm{x_b^t}) \qquad (11)$$

Where $rand_3$ and $rand_4$ are normally distributed stochastic random numbers in the range [0, 1].

The Pareto based Cuckoo search algorithm pseudo code is shown in *Figure 2*.

```
Objective function $(fx)$, $x = (x_1, x_2 \ldots\ldots x_d)^T$
Generate initial population of n host nests $xi$ ($i = 1,2..n$)
**While** (t < Max Generation) or (stop criteria)
   Get a cuckoo (say $i$ ) randomly by Pareto 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 ($P_a$ ) of worse nests are abandoned and
   new ones are built at new locations via Levy flights;
   Keep the best solutions (or nests with quality solutions);
   Rank the solutions and find the current best;
**End while**
Post processing
```

Figure 2 Pareto based CS pseudo code for a global optimisation

4.Methodology

4.1Fair comparison

Fair comparison of optimisation algorithms required a lot of thought before one can conclude which algorithm is better than the other. The main challenge is that during the optimisation process, algorithms do not start at the same point. The second challenge is that the algorithms follow different paths due to the nature of stochastic randomness. These problems are addressed by ensuring firstly that the algorithms are tested on the same test functions or problems. Then the critical solution to run simulations for a defined number of times and then determine some statistical variables like maximum, mean, minimum and standard deviation values for each algorithm. The statistical results provide a fair indication based on the test problems only as to which algorithms performed better. On the other hand, then if algorithm A performs better than algorithm B on one test function, it does not mean that algorithm A can perform better than algorithm B for all test functions or problems[19]. No-free-lunch theorem states that there is no algorithm that can outperform all other algorithms for all the problems [20].

4.2Benchmarking and measure

There is no optimiser that can be classified as best for all problems. That means the optimizer can only be claimed to be best for the test functions it has been tested and it would be incorrect to generalise best on few test functions used. Benchmarking is a process of determining an algorithm's capability to optimise a group of problems using systematic tests. Normally, a set of test problems is available and there is no need to recompile these test functions [19].

4.3Characteristics of test functions

Ten benchmark test functions are introduced in this section. Some of these functions are unimodal, have only one optimum position and other functions are multimodal, have more than one local optimum position.

Ackley function is a multimodal continuous test function mostly used in testing minimisation optimisation algorithms. It has large number of local minima and only one global minimum[21]. It is defines by (12).

$$f(x) = -20\exp[-\left(\frac{1}{5}\right)[\left(\frac{1}{n}\right)\sum_{i=1}^{n} x_i^2]^{0.5} - \exp[\left(\frac{1}{n}\right)\sum_{i=1}^{n}\cos(2\pi\ x_i)] + 20 + e \qquad (12)$$

Where, n = 1,2., $-32.768 \leq x_i \leq 32.768$ and i = 1,2,..,n.

This has a global minimum $f_* = 0$ at $x_* = (0,0,.0)$.

Griewank function is a multimodal function with many widely and regularly spread local minima. Its number of minima grows exponentially when the number of dimensions increases [21]. It is defined by (13).

$$f(x) = \left(\frac{1}{4000}\right)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} co\,s\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad (13)$$

Where, $-600 \leq x_i \leq 600$ and the global minimum $f_* = 0$ at $x_* = (0,0,..0)$.

Bohachevsky function [9] is a multimodal minimisation test function defined in (14).

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi\ x_1) - 0.4\cos(4\pi\ x_2) + 0.7 \qquad (14)$$

Where, $-100 \leq x_i \leq 100$ and the global minimum $f_* = 0$ at $x_* = (0,0)$.

De Jong function is a unimodal, convex, simple and widely used test function with no local but one global minimum [22].

$$f(x) = \sum_{i=1}^{n} x_i^2 \qquad (15)$$

Where, $-5.12 \leq x_i \leq 5.12$ and the global minimum $f_* = 0$ at $x_* = (0,0,..0)$.

Matyas function [9] is a unimodal defined as;

$$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2 \qquad (16)$$

Where, $-10 \leq x_i \leq 10$ and the global minimum is located at $x_*= (0,0)$ with $f_* =0$.

Zakharov function is a unimodal test function with only one global minimum but no local minima [9].

$$f(x) = \sum_{i=1}^{n} x_i^2 + (0.5\sum_i^n ix_i)^2 + (0.5\sum_i^n ix_i)^4 \qquad (17)$$

Where, $-5 \leq x_i \leq 10$ and the global minimum is $f_* = 0$ at $x_* = (0, 0,..0)$.

Goldstein-Prices function is a slightly multi-modal continuous test function with no local minima but one global minimum [9].

$$f(x) = 1 + (x_1 + x_2\ 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2\ 3x_2^2)] * [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 32x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \qquad (18)$$

Where, $-2 \leq x_i \leq 2$ and the global minimum $f_* = 3$ at $x_* = (0,-1)$.

Rosenbrock function is a unimodal test function with one global minimum in a narrow parabolic valley and no local minima. However, it is difficult to find the global minima since the valley is very non-linear. Optimisation algorithms can converge slowly because they constantly have to change directions [23].

$$f(x) = \sum_{i=1}^{n-1}[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2] \qquad (19)$$

Where, i = 1,2...,n and $-5 \leq x_i \leq 5$ and has global minimum $f_* = 0$ at $x_* = (1,1,...1)$.

Easom function is a unimodal test function with several local minima and a global minimum found in a very small area relative to the entire search area. [9].

$$f(x) = -\cos(x)\,Cos(y)\exp[-(x - \pi)^2 + (y - \pi)^2] \qquad (20)$$

Where, n = 1,2., and $-100 \leq x,y \leq 100$ and has a global minimum $f_* = -1$ at $x_* = (\pi,\pi)$.

Michalewicz function [24] is a highly multimodal test function with n! local minima .

$$f(x) = -\sum_{i=1}^{n} \sin(x_i) . [\sin(\frac{i x_i^2}{\pi})]^{2m} \qquad (21)$$

Where, m = 10 (default), and $0 \leq x_i \leq \pi$ for i= 1, 2….n. and has a global minimum $f_* = -9.66$.

4.4Simulation settings

Figure 3 depicts methodology flowchart. Five CS algorithms (Levy, Cauchy, Gauss, Gamma and Pareto) were developed in MATLAB version 7.10.0.499 (R201a). The number of Cuckoo nests (population size n) was fixed at n =25. The probability of discovering an egg (P_a) was fixed at 0.25. Each test function was run 20 times and statistical variables (maximum, minimum, mean and standard deviation) were determines in order to overcome the issue of random walk nature of CS algorithms. The average computational time for each CS version was also recorded.

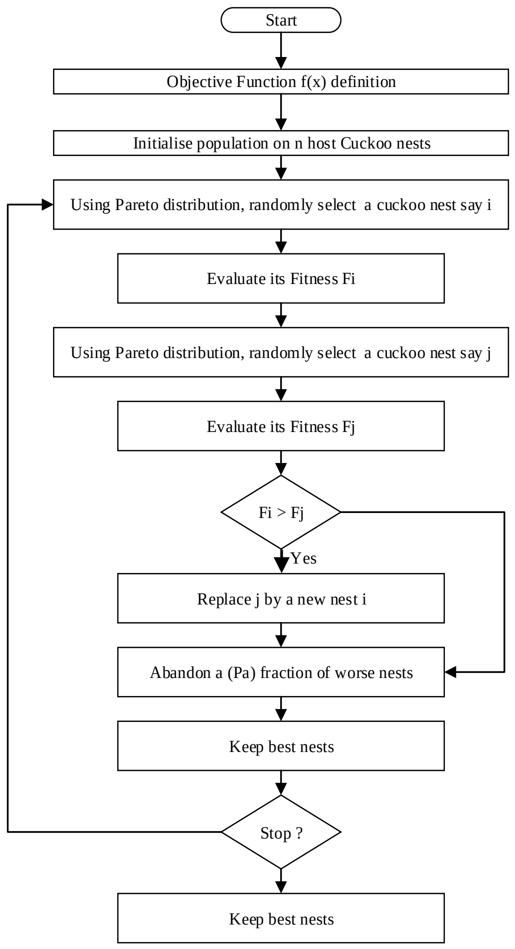
Start
Objective Function f(x) definition
Initialise population on n host Cuckoo nests
Using Pareto distribution, randomly select a cuckoo nest say i
Evaluate its Fitness Fi
Using Pareto distribution, randomly select a cuckoo nest say j
Evaluate its Fitness Fj
Fi > Fj
Yes
Replace j by a new nest i
Abandon a (Pa) fraction of worse nests
Keep best nests
Stop ?
Keep best nests

Figure 3 Proposed methodology flowchart

The test environment was FUJITSU laptop with the following specifications; RAM: 3.0GB, CPU: Intel Celeron 900@2.2Ghz and 32bit windows 7 home Basic operating system.

5.Results and discussions

This section presents and discusses the results on the four CS versions; namely the Levy, Gauss, Cauchy, Gamma and Pareto based CS algorithms.

From *Table 1*, Pareto outperformed other CS algorithms in three out of ten test functions, namely; Matyas, Zakharov and Rosenbrock. The next algorithm is Cauchy which outperformed all the algorithms in two test functions (namely; Griewank and De Jong). Levy-based CS outperformed all other algorithms in one test function, Michalewicz. For Ackley test functions, the Levy, Cauchy and Pareto performed equally well followed by Gauss and then Gamma. All the five algorithms performed the same for Bohachevsky, Goldstein-Prices and Easom test functions.

From *Figure 4*, the order of the CS algorithms convergences is as follows; The Pareto-based CS algorithm is approaching the global minima value of zero faster than the other CS algorithms. Then the Levy based CS algorithm follows, the third algorithm is Cauchy-based CS. The fourth position is Gauss and the last is Gamma based CS. The information obtained from *Figure 5* shows that Pareto based CS is the fastest algorithm to converge towards the value of zero. The convergence rate of other versions on CS are as follows; Levy, Gauss, Cauchy and Gamma.

Pareto based CS took about 10 iterations to converge to zero while Levy based CS took as per *Figure 6*. The Cauchy, Gauss and Gamma took 38, 42 and 45 iterations, respectively.

The Pareto based CS converged to zero after 52 iterations in *Figure 7*. The other algorithms had not reached the correct value of zero after 100 iterations. However, their convergence order was Cauchy lead, followed by Levy and Gamma, with Gauss is the last or slow to converge.

In *Figure 8* the order of convergence rate starting from fastest to slowest is as follows; Cauchy, Pareto, Gamma, Levy and Gauss. All functions converged to the correct value of zero in less than 50 iterations.

The plots of *Figure 9* shown that Pareto based CS outperformed other CS algorithm (Gauss, Levy, Cauchy and Gamma) by converging quickly to the true value of zero. Pareto based CS converged very quick after 10 iterations to correct value of zero. The other algorithms (Levy, Gamma, Gauss and Cauchy) had not reached correct value after 100 iterations due to their slowly convergence from *Figure 11*.

Table 1 Global minima values

Function	Parameters	Levy	Cauchy	Gauss	Gamma	Pareto
Ackley	Max	4.45E-15	4.44E-15	7.99E-15	7.99E-15	4.45E-15
	Min	4.45E-15	4.44E-15	4.44E-15	4.44E-15	4.45E-15
	Mean	4.45E-15	4.44E-15	4.97E-15	4.97E-15	4.45E-15
	Sd	0	0	1.302E-15	1.302E-15	0
	Duration	1.51E-3s	1.21E-03s	1.22E-03s	4.42E-03s	1.27E-3s
Griewank	Max	5.929E-14	0	5.918E-14	4.030E-11	4.241E-14
	Min	0	0	0	0	0
	Mean	3.103E-15	0	4.741E-15	2.365E-15	2.132E-15
	Sd	1.323E-14	0	1.483E-14	9.039E-15	9.481E-15
	Duration	1.48E-3s	1.11E-03s	1.04E-03s	2.04E-03s	1.09E-3s
Bohachevsky	Max	0	0	0	0	0
	Min	0	0	0	0	0
	Mean	0	0	0	0	0
	Sd	0	0	0	0	0
	Duration	1.22E-3s	7.88E-04s	8.47E-04s	1.68E-03s	7.94E-4s
De Jong	Max	4.33E-35	1.27E-36	4.38E-33	4.87E-33	3.12E-36
	Min	4.19E-37	3.40E-39	6.47E-35	2.12E-35	6.05E-38
	Mean	1.17E-35	1.43E-37	5.68E-34	9.39E-34	9.49E-37
	Sd	1.301E-35	3.15338E-37	9.723E-34	1.131E-33	8.755E-37
	Duration	1.19E-3s	8.43E-04s	9.54E-04s	1.72E-03s	7.77E-4s
Matyas	Max	4.05E-66	6.04E-73	9.35E-68	6.89E-96	3.14E-74
	Min	4.18E-79	7.26E-86	3.09E-81	7.35E-79	3.03E-83
	Mean	2.81E-67	3.04E-74	4.74E-69	5.93E-70	2.62E-75
	Sd	9.5382-67	1.34997E-73	2.089E-68	1.672E-69	8.05E-75
	Duration	1.20E-3s	7.66E-04s	7.87E-04s	1.67E-03s	7.90E-4s
Zakharov	Max	1.96E-20	4.77E-21	5.08E-20	1.23E-19	1.08E-21
	Min	3.35E-22	1.88E-23	1.46E-21	2.15E-21	4.26E-23
	Mean	4.98E-21	7.12E-22	1.46E-20	4.26E-20	4.13E-22
	Sd	4.829E-21	1.20198E-21	1.138E-20	3.209E-20	3.213E-22
	Duration	1.20E-3s	7.81E-4s	7.96E-04s	1.67E-03s	8.63E-4s
Goldstein-Prices	Max	3	3	3	3	3
	Min	3	3	3	3	3
	Mean	3	3	3	3	3
	Sd	0	0	0	0	0
	Duration	1.23E-3s	7.86E-04s	7.91E-04s	1.68E-03s	7.54E-4s
Rosenbrock	Max	4.35E-4	2.59E-04	1.48E-04	3.25E-04	6.08E-5
	Min	2.46E-9	2.61E-08	1.90E-08	1.53E-09	8.39E-9
	Mean	2.66E-5	3.13E-05	1.26E-05	4.51E-05	4.55E-6
	Sd	9.7262E-5	7.54288E-05	3.401E-05	8.858E-05	1.3873E-5
	Duration	1.23E-3s	7.82E-04s	7.69E-04s	1.64E-03s	7.77E-4s
Easom	Max	-1	-1	-1	-1	-1
	Min	-1	-1	-1	-1	-1
	Mean	-1	-1	-1	-1	-1
	Sd	0	0	0	0	0
	Duration	1.48E-3s	8.05E-04s	8.00E-04s	1.83E-03s	8.01E-4s
Michalewicz	Max	-7.56	-7.25	-7.45	-7.21	-7.2
	Min	-9.08	-9.26	-8.81	-8.51	-8.52
	Mean	-8.39	-7.89	-7.99	-7.95	-7.93
	Sd	0.402096	-0.444263	0.349764	-0.374807	0.360561
	Duration	1.62E-3s	1.21E-03s	1.42E-03s	2.06E-03s	1.70E-3s

From *Figure12*, the Cauchy based CS algorithm is the first to converge to the correct value of -1. The second algorithm to converge to correct value is Gauss based CS, and then Gamma based CS and then the Pareto. Finally the last algorithm to converge to correct value is Levy based CS.

From the *Figure 13*, all other CS based algorithms except the Pareto are converging to values below the -9.66 which is incorrect since the global minimal value of Michalewicz is -9.66. In this case, the Pareto-based CS algorithm is considered the best, since it is still converging and it has not passed the -9.66 value.

The general observation is that Pareto outperformed other CS algorithms in 6 test functions as per *Figure 4, 5, 6, 7, 9* and *11*, out of *10* test functions in converging closest to the global minimum value. The second best performing algorithm is Cauchy CS outperformed in three test functions as per *Figure 8, 10 and 12*.

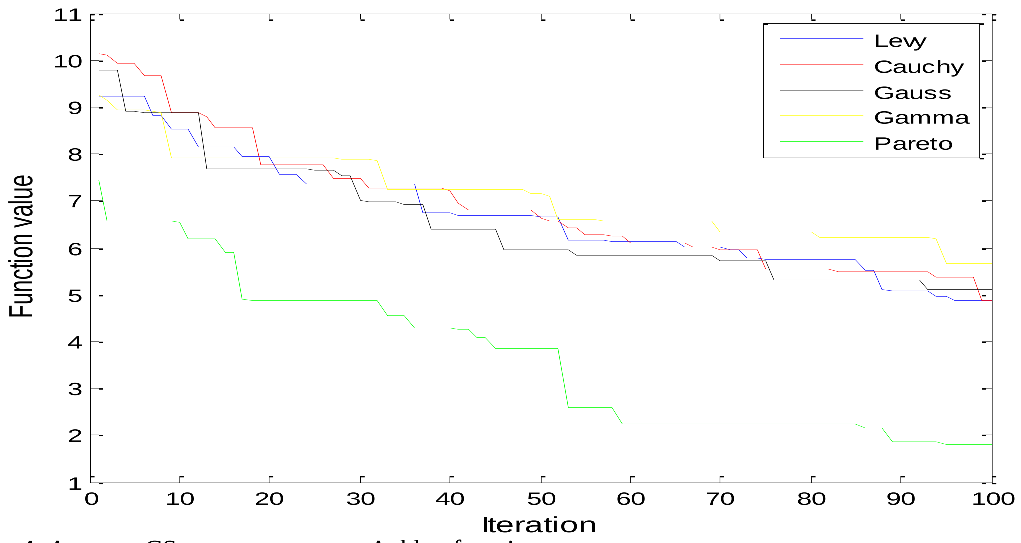

Figure 4 Average CS convergences on Ackley function

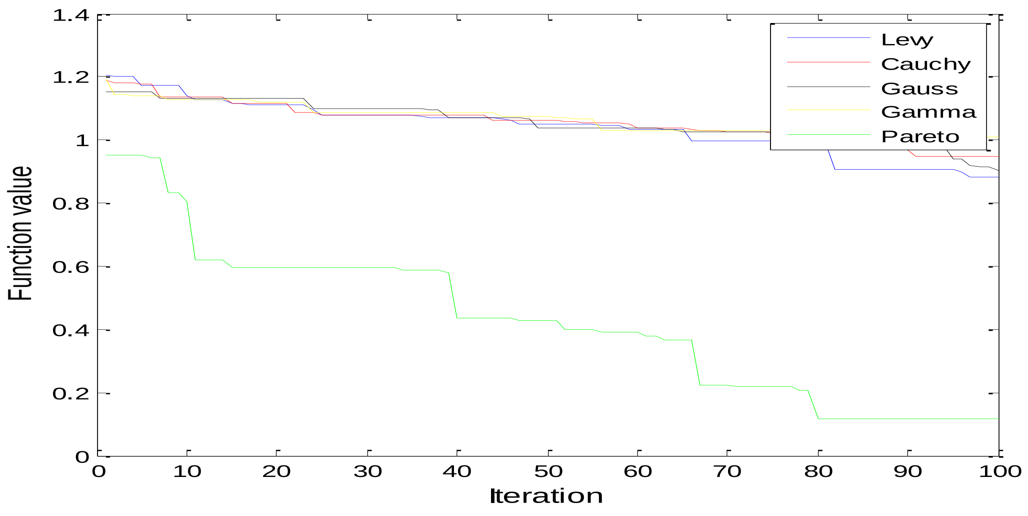

Figure 5 Average CS convergences on Griewank function

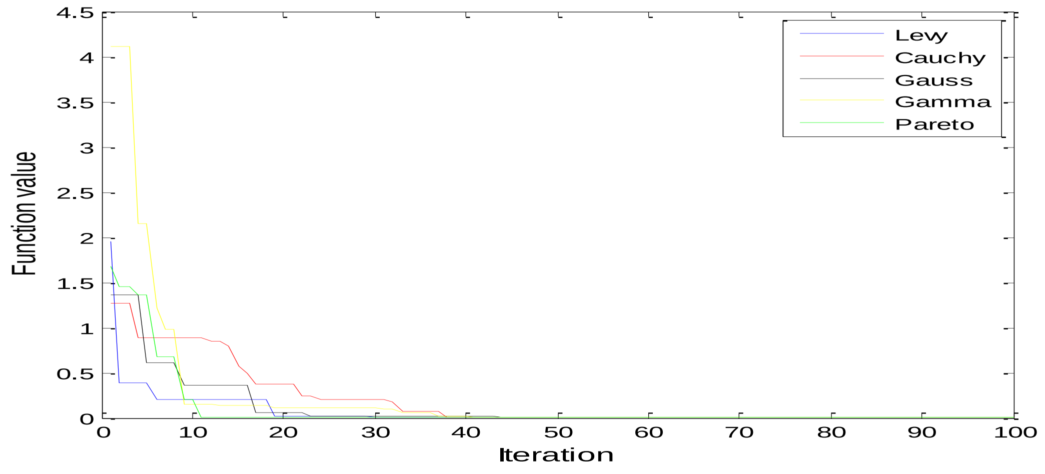

Figure 6 Average CS convergences on Bohachevsky function

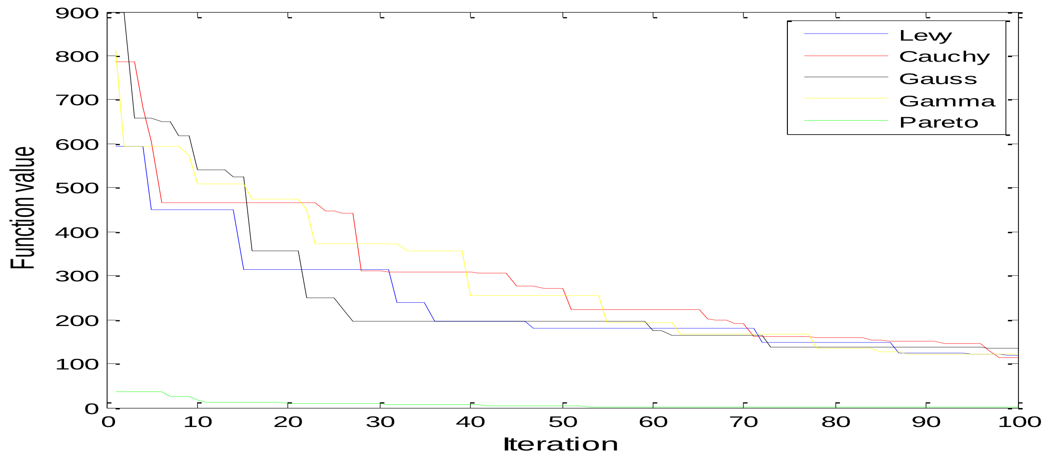

Figure 7 Average CS convergences on De Jong function

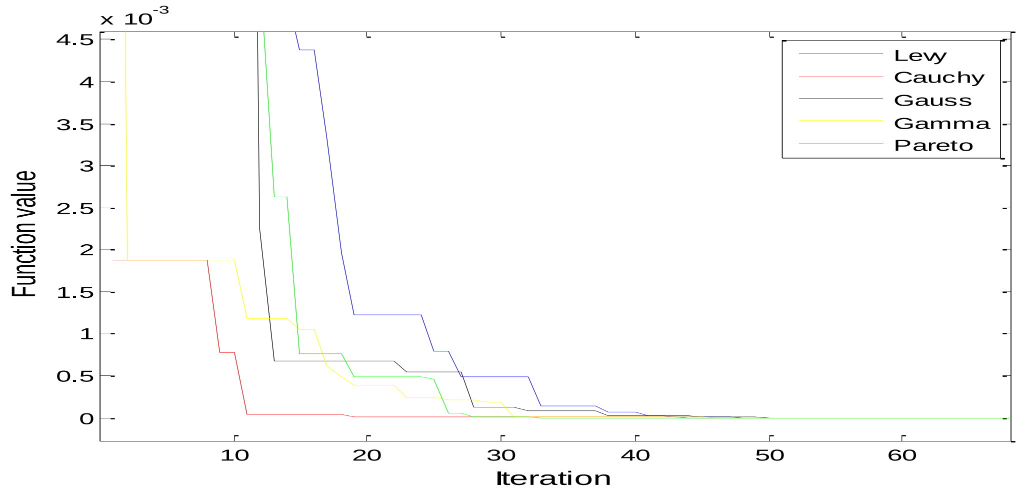

Figure 8 Average CS convergences on Matyas function

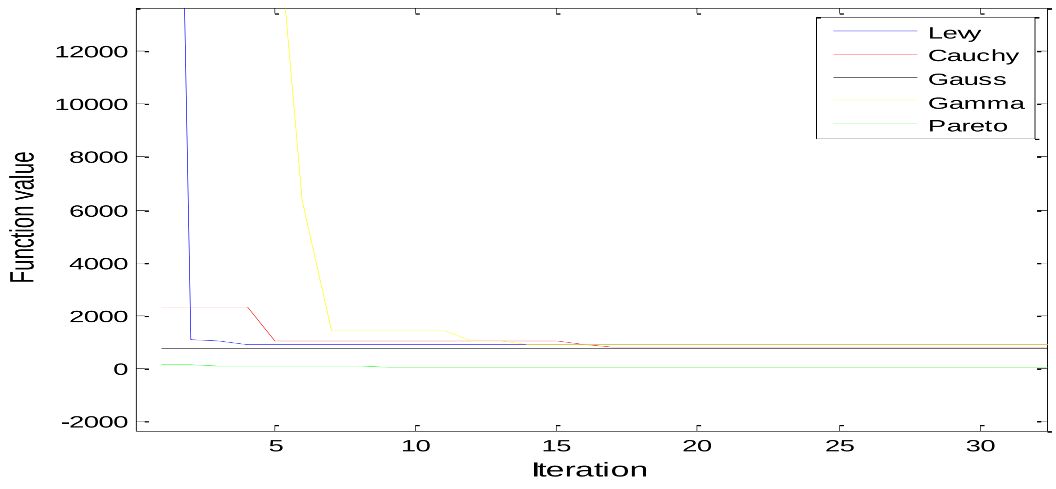

Figure 9 Average CS convergences on Zakharov function

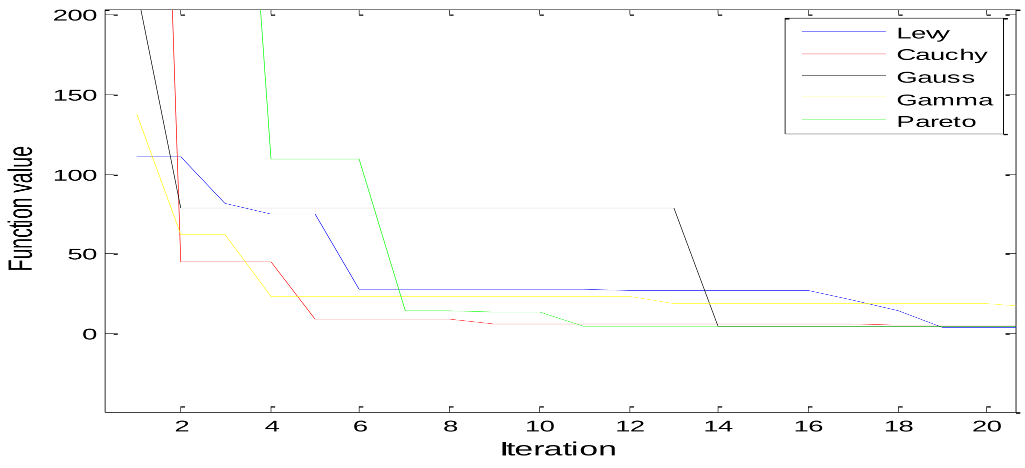

Figure 10 Average CS convergences on Goldstein-Price function

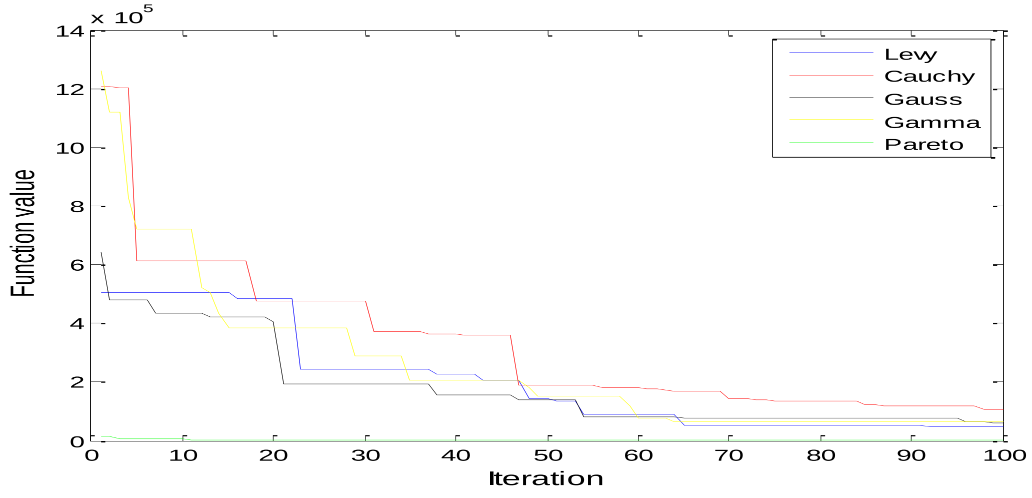

Figure 11 Average CS convergences on Rosenbrock function

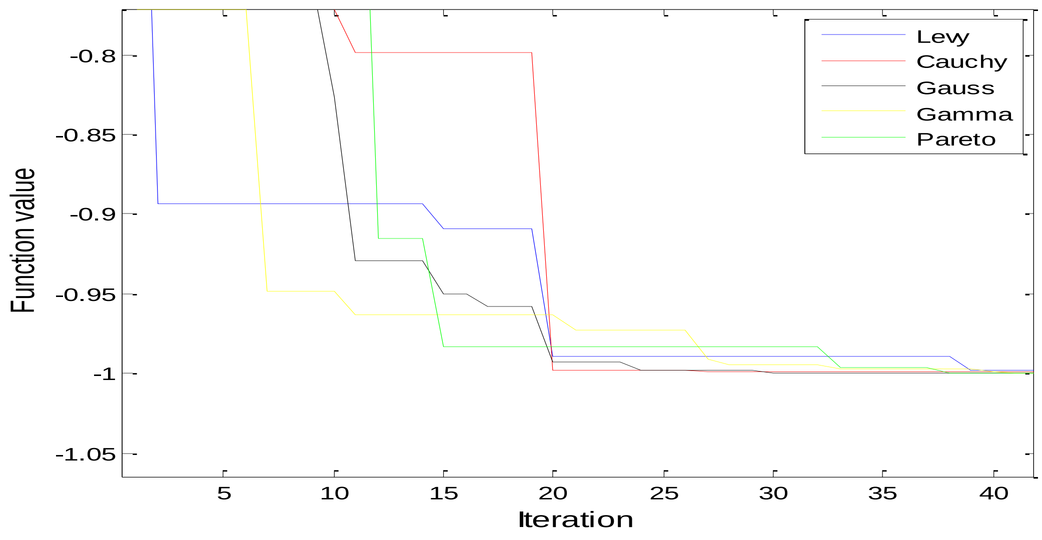

Figure 12 Average CS convergences on Easom function

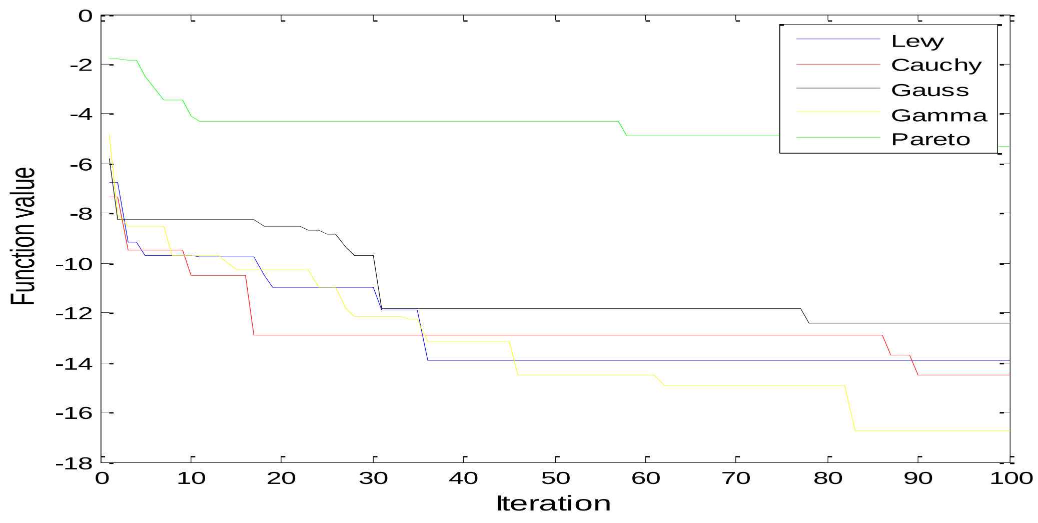

Figure 13 Average CS convergences on Michalewicz function

6.Conclusions and future work

This paper implemented a new CS algorithm whose step size is derived from Pareto distribution. Its performance was validated using standard test functions and compared to CS algorithms with step sizes determined from other probability distributions. In terms of accuracy,

The general observation is that Pareto outperformed other CS algorithms in 6 test functions as per *Figure 4, 5, 6, 7, 9 and 11, out of 10* test functions in converging closest to the global minimum value. The second best performing algorithm is Cauchy CS outperformed in three test functions as per *Figure 8, 10 and 12*.

One of future works where the Pareto based cuckoo search algorithm can be used is to train neural networks for electricity load forecasting.

Acknowledgment

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 Optimization Algorithms and also to the anonymous reviewers for their valuable comments.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Belič E, Lukač N, Deželak K, Žalik B, Štumberger G. GPU-based online optimization of low voltage distribution network operation. IEEE Transactions on Smart Grid. 2017; 8(3):1460-8.

[2] Koppel A, Sadler BM, Ribeiro A. Proximity without consensus in online multi-agent optimization. In IEEE international conference on acoustics, speech and signal processing 2016 (pp. 3726-30). IEEE.

[3] Platonov A. Information theory and optimization of analog feedback communication systems. In black sea conference on communications and networking (BlackSeaCom) 2016 (pp. 1-5). IEEE.

[4] Tang M, Gao L, Pang H, Huang J, Sun L. Optimizations and economics of crowdsourced mobile streaming. IEEE Communications Magazine. 2017; 55(4):21-7.

[5] Das S, Doppa JR, Pande PP, Chakrabarty K. Design-space exploration and optimization of an energy-efficient and reliable 3-D Small-world network-on-chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2017; 36(5):719-32.

[6] Yan Z, Fan J, Wang J. A collective neurodynamic approach to constrained global optimization. IEEE Transactions on Neural Networks and Learning Systems. 2017; 28(5):1206-15.

[7] Parkinson AR, Balling R, Hedengren JD. Optimization methods for engineering design. Brigham Young University. 2013.

[8] Hegazy O, Soliman OS, Salam MA. Comparative study between FPA, BA, MCS, ABC, and PSO algorithms in training and optimizing of LS-SVM for stock market prediction. International Journal of Advanced Computer Research. 2015; 5(18):35-45.

[9] Yang XS. Nature-inspired optimization algorithms. Elsevier. 2014.

[10] Zheng H, Zhou Y. A novel cuckoo search optimization algorithm based on Gauss distribution. Journal of Computational Information Systems. 2012; 8(10):4193-200.

[11] Zaw MM, Mon EE. Web document clustering using gauss distribution based cuckoo search clustering algorithm. International Journal of Scientific Engineering and Technology Research. 2014;3(13): 2945-9.

[12] Ho SD, Vo VS, Le TM, Nguyen TT. Economic emission load dispatch with multiple fuel options using cuckoo search algorithm with Gaussian and Cauchy distributions. International Journal of Energy, Information and Communications. 2014; 5(5): 39-54.

[13] Nguyen TT, Vo DN, Dinh BH. Cuckoo search algorithm using different distributions for short-term hydrothermal scheduling with reservoir volume constraint. International Journal on Electrical Engineering and Informatics. 2016; 8(1):76-92.

[14] Roy S, Mallick A, Chowdhury SS, Roy S. A novel approach on Cuckoo search algorithm using Gamma distribution. In international conference on electronics and communication systems 2015 (pp. 466-8). IEEE.

[15] Yang XS, Deb S. Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation. 2010; 1(4):330-43.

[16] www.albany.edu/~hammond/gellmu/examples/gamma.pdf. Accessed 19 May 2016.

[17] Raja TA, Mir AH. On fitting of generalized Pareto distribution. Global Journal of Human-Social Science Research. 2013; 13(2):8-11.

[18] http://www.math.uah.edu/stat/special/Pareto.html. Accessed 15 November 2016.

[19] Opara K, Arabas J. Benchmarking procedures for continuous optimization algorithms. Journal of Telecommunications and Information Technology. 2011:73-80.

[20] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1997; 1(1):67-82.

[21] Yang XS. Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons; 2010.

[22] Nagham Azmi AM, Khader AT. De Jong's sphere model test for a social-based genetic algorithm (SBGA). International Journal of Computer Science and Network Security (IJCSNS). 2008; 8(3):179-85.

[23] http://www.redcedartech.com. Accessed 04 November 2016.

[24] www.robertmarks.org. Accessed 29 April 2017.

Mahlaku Mareli received BSc.Eng (Electrical and Eelctronic) Engineering from University of Cape Town in 1996. In 2014 he obtained MEng (Electrical and Electronic) Engineering from University of Johannesburg. He is currently a PhD student at the University of Johannesburg. His research interests are in optimisation, Neural Networks and Artificial Intelligence.

Email: mmareli@hotmail.com

Bheki Twala received MSc (Statistics) in 2005 from the University of Southampton and PhD (Machine Learning and Statistical Science) from Open University, UK. He did his postdoctoral research at Brunel University, UK. He is Director of Institute for Intelligent Systems at University of Johannesburg and before that he was Head of Department of Electrical and Electronic Engineering Science at University of Johannesburg. In 2016 he won prestigious annual TW Kambule NSTF research award for his work in building on diverse experience in using Artificial Intelligence to solve several problems in many industries.