Enhancing Modified Cuckoo Search Algorithm by using MCMC Random Walk

Noor Aida Husaini, Rozaida Ghazali
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia (UTHM),
86400 Parit Raja, Batu Pahat, Johor.
gi090003@siswa.uthm.edu.my, rozaida@uthm.edu.my

Iwan Tri Riyadi Yanto
Department of Information System,
Ahmad Dahlan University,
Yogyakarta, Indonesia.
yanto.itr@is.uad.ac.id

***Abstract*— In this paper, we scrutinised an improvement of the Modified Cuckoo Search (MCS), called Modified Cuckoo Search-Markov chain Monte Carlo (MCS-MCMC) algorithm, for solving optimisation problems. The performance of MCS are at least on a par with the standard Cuckoo Search (CS) in terms of high rate of convergence when dealing with true global minimum, although at high number of dimensions. In conjunction with the benefits of MCS, we aim to enhance the MCS algorithm by applying Markov chain Monte Carlo (MCMC) random walk. We validated the proposed algorithm alongside several test functions and later on, we compare its performance with those of MCS-Lévy algorithm. The capability of the MCS-MCMC algorithm in yielding good results is considered as a solution to deal with the downside of those existing algorithm.**

Keywords*— *MCS-MCMC, modified cuckoo search, cuckoo search, Markov chain Monte Carlo

I. INTRODUCTION

Metaheuristics have been established as one of the most practical approach to optimisation [1-7]. These intelligent mechanisms, which include Cuckoo Search (CS), offers great advantages over conventional modeling, including the capabilities to employs high level techniques in exploring and exploiting the search space. CS is a new evolutionary optimisation algorithm which was influenced by the obligate brood parasitism of some cuckoo species in collaboration with the Lévy flight behaviour of some birds and fruit flies [8]. They laid their egg in neighbourhood nests, ever since they do not have their own nests. In order to upsurge the hatching possibility, they may remove the eggs in the nests of other host birds and mimicry the eggs of those host birds [8]. Therefore, in consideration of finding the best environment for breeding and reproduction, the cuckoos' group tends to immigrate. Due to its substantial theoretical importance, that CS algorithm has been explored in the context of simplicity due to fewer parameter settings and quick convergence speed, which makes the algorithm inclusively and indomitably used for many optimisation problems [1, 3, 4, 9-11].

Numerous improvements being made, in consideration of magnifying the accuracy and the rate of convergence of this algorithm. For instance, Valian et al. [12] introduced an Improved Cuckoo Search (ICS) algorithm by adjusting the parameters, probability P_α and step size α. Those parameters are important to fine-tuning the solution vectors. To magnify the accuracy and the rate of convergence, Walton et al. [13] modified the standard CS algorithm by adding up the information exchange between the top eggs. The Modified Cuckoo Search's (MCS) performances are at least on a par with the standard CS in terms of high rate of convergence to the true global minimum remarkably at high number of dimensions. In conjunction with the benefits of MCS, we inclined an initiative to enhance the MCS by replacing the Lévy flight found in the algorithm with Markov chain Monte Carlo (MCMC) random walk. A major advantage of the MCMC theory is that, it learns better parameter automatically whilst ruling good parameter values by little user intercession. Those can be achieved by Markov chain mixing and integrated autocorrelation of a functional of interest.

The paper is mainly organised according to five (5) sections. In the first section, a brief capsulation of the study is disposed. In the second part, the key concept behind the MCS with pseudo code is discussed. The third section comprises the proposed MCS-MCMC algorithm. In the fourth section, the experimental setup is clarified and the experimental results are presented. In the fifth section which is the last section, some conclusions and future work are untaken.

II. MODIFIED CUCKOO SEARCH

According to Yang and Deb [8], the CS will always find the optimum if it been given enough computation. Basically, the search technique in CS being done by considering the whole area on random walks. However, it is not guaranteed whether the exploration can converged faster or not. Typically, the step size α in CS are kept constant which resulting the efficiency of the algorithm to tail off. To contend with this concern, Walton et al. [13] created two (2) modifications: 1) change the step size of Lévy flight, α. In ordinary CS, the value of α was constantly kept by employing $\alpha = 1$ [8]. 2) hasten up the rate of convergence by taking into account the information exchange amongst the eggs. Since the exploration in the ordinary CS is performed by their own selves, therefore, the information exchange amongst the eggs is not exist. As for the MCS, the eggs were evaluated twice. The first evaluation involves putting a sub of the eggs upon the best fitness into a

group of top eggs. Apiece of the top eggs, a second egg in this group is evaluated by picking up randomly before a new egg being generated on the line that connects these two top eggs. With regard to get the best fitness, the new location of the new egg (distance along the line) is determined by employing the inverse of the golden ratio $\varphi = \left(1+\sqrt{5}\right)/2$. If the same fitness value found in both eggs, the new egg is generated at the center point.

There are 2 parameters that need to be adjusted in the MCS, which refer to the fraction of nests to be abandoned and the fraction of nests to generate the top nests. The initial value of the step size of Lévy flight $A=1$ is chosen. At each generation, a new step of Lévy flight is calculated by using $\alpha \leftarrow A/\sqrt{G}$, where G specifies the number of generation. This exploration searching is only can be used for the fraction of nests to be abandoned. There is a probability that, in this measurement, the same egg is chosen twice. Therefore, by performing a local Lévy flight search on the randomly picked nest with step size $\alpha \leftarrow A/G^2$ can handle these problems. The step-by-step processes in the MCS are presented in Algorithm 1.

$A \leftarrow MaxLevyStepSize$
$\varphi \leftarrow GoldenRatio$
Initialise a population of n nests $x_i (i = 1,2,\ldots,n)$
FOR $\forall x_i$, do
    Calculate fitness $F_i = f(x_k)$
ENDFOR
Generation number $G \leftarrow 1$
WHILE
NumberObjectiveEvaluations < *MaxNumberEvaluations*, do
    $G = G + 1$
    Sort nests by order of fitness
    FOR all nests to be abandoned, do
        Arrange nests by fitness from recent position x_i
        Compute the step size of Lévy flight $\alpha \leftarrow A/\sqrt{G}$
        Perform Lévy flight from x_i to generate new egg x_k
        $x_i \leftarrow x_k$
        $F_i \leftarrow f(x_k)$
    ENDFOR
    FOR all of the top nests do
        Recent position x_i
        Randomly select new nest from the top nests x_j
        IF $x_i = x_j$ then
            Calculate the step size of Lévy flight $\alpha \leftarrow A/G^2$
            Execute Lévy flight from x_i, to produce new egg
            x_k, $F_k = f(x_k)$
            From all nests, randomly select nest l
            IF $(F_k > F_l)$ do
                $x_l \leftarrow x_k$
                $F_l \leftarrow F_k$
            ENDIF
        ELSE
            $dx = |x_i - x_j|/\varphi$
            Move the worst nest distance dx to the
            best nest in order to find $x_k, F_k = f(x_k)$
            Randomly select nest l from all nests
            IF $(F_k > F_l)$ then
                $x_l \leftarrow x_k$
                $F_l \leftarrow F_k$
            ENDIF
        ENDIF
    ENDFOR
ENDWHILE

ALGORITHM 1: Modified Cuckoo Search (MCS)

III. RANDOM WALK

In 1905, Karl Pearson proposed the term “random walk”. The random walk is a path that considers the mathematical formalisation that comprises a sequence of random steps [14]. It is noted that the random walk is presumed to be uniform, symmetric, intricate, and possess zero mean and delimited variance of jumps [14]. The areas of random walk includes the field of ecology [15], economics [16], computer science [17] and the like. Commonly, the random walks work on graph, while some are on the line. Additionally, it can be found on the plane, or in high dimensions, or on groups. Typically, animal hunting for food in a random direction. The chosen direction is depending indirectly on a probability of both the current location/state and the transition probability to the next location.

A. Lévy Flight

A Lévy flight is a type of random walk that the distribution of the step-lengths lies within a heavy-tailed probability distribution. The typical properties of this kind of distribution is the positive exponential moments are infinite (do not have finite mean and variance). The term “Lévy flight” was invented by Benoît Mandelbrot [18] who used this for one specific definition of the distribution of step sizes. If the distribution of step size is a Cauchy distribution, he used the term, “Cauchy flight”. If the distribution is a normal distribution, he used the term, “Rayleigh flight”. For a particular case, when the directions of the step sizes are in isotropic random directions, he used the term, “Lévy flight” [18] which is defined by the survivor function of the distribution of step-sizes, U, as in (1):

$$\Pr(\mathrm{U} > \mathrm{u}) = \begin{cases} 1 & : \ u < 1, \\ u^{-D} & : \ u \geq 1. \end{cases} \quad (1)$$

where D is a parameter in the fractal dimension that lies in Pareto distribution.

Theoretical research has shown that the distribution of step sizes can be any distribution for which the Lévy exponent of approximately 2 (also called power law) can provide a higher efficiency than other exponents:

$$PR(U > u) = O(u^{-\beta}) \tag{2}$$

that fulfill $1 < \beta < 3$ [18].

B. Markov chain Monte Carlo

MCMC methods are constructed based on a Markov chain which has the desired distribution on account of its equilibrium distribution. Due to the aptitude of MCMC that provides internal into broad, transcendent Bayesian issues, it has been used as one of the utmost significant reinforcement in modern statistics [19]. For sampling the desired distribution, the MCMC used the chain state after an adequate number of steps. The quality sample is used for improving the function of the number of steps. Nevertheless, the issues on how to define and minimising the steps required for it to fully converge into equilibrium distribution remains problems by researchers. The distribution is said to be in equilibrium state when the process develop gradually in a random way until it achieves a certain point, that remains as a subsequently distribution [19]. As there is always some issue regarding the starting position, the MCMC sampling can only approximate the target distribution.

The MCMC sampling regularly used in calculating multi-dimensional integrals numerically, whereas a group of "walkers" moves around at random. The integrand value at each point where the walker steps, is calculated close to the integral. Further, a pile of tentative steps encompassing the area is taken, wherever a place with apparently have huge contribution to the integral being chose so that later on, they may move into the next step [19].

IV. The Proposed MCS-MCMC Algorithm

The motivation of using MCMC in the MCS algorithm is due to the fact that the MCMC performs full-dimensional jumps at each iteration. It also has a large-scale polynomial rate of convergence due to the presence of central limit theorem (CLT) for higher moment. The MCS-MCMC algorithm generally can be depicted in a flowchart as given in Fig. 1.

The MCMC is involved 2 parts which was circled in red (Refer to Fig. 1). In this step, we apply the MCMC random walk due to the benefits: higher polynomial convergence rate due to the presence of CLT for higher moment. This is because, MCMC use local moves based on certain kinds of target density prior to better algorithms qualitatively.

The MCS-MCMC is operated by the following steps:

- At first, we generate the initial value, θ that satisfies $f(\theta) > 0$, by considering the target probability density function (PDF):

$$p(\theta) = C.\theta^{-n/2}.\exp\left(\frac{-a}{2\theta}\right) \tag{3}$$

  where $n = 5$ and $a = 4$.

- There are 2 parameters, number of samples (iterations) and samples drawn from the target PDF, $p(\theta)$. For the number of samples, we set the value as the same as the dimension of each test function. Then, we marked-out U from a uniform distribution at random, and accept θ subject to $U \le P$ where $P < 1, \theta$.

- For the procedure, we calculate the density ratio at the candidate point, θ and current points, θ_{t-1},

$$P = \frac{p(\theta)}{p(\theta_{t-1})} = \frac{f(\theta)}{f(\theta_{t-1})} \tag{4}$$

- We can summarise that the sampling as first computing, and then accept the probability, P

$$P = \min \frac{f(\theta)}{f(\theta_{t-1})}, 1 \tag{5}$$

A. Test Functions

In the literature [20, 21], numerous benchmark test functions has been designed to assess the performance of optimisation algorithms. With regard to verify the proficiency of the proposed MCS-MCMC algorithm, we used three (3) benchmark test functions [20, 21]:

Ackley's Function is multimodal. It is broadly used for testing the optimisation algorithms [22].

$$f(x) = -20\exp\left[-0.2\sqrt{\left(\frac{1}{d}\sum_{i=2}^{d} x_i^2\right)}\right] - \exp\left[\frac{1}{d}\sum_{i=2}^{d}\cos(2\pi x_i)\right] + (20 + e) \tag{6}$$

with a global minima $f^* = 0$ at $x^* = (0,0,\ldots,0)$ in the range of $-32.768 \le x_i \le 32.768$ where $i = 1,2,\ldots,d$.

Rosenbrock's Function is unimodal. The global minima $f(x^*) = 0$ at $x^* = (1,1,\ldots,1)$ between $[-100,100]$ is enclosed by a long, narrow, parabolic shaped flat valley [23].

$$f(x) = \sum_{i=1}^{d-1}\left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2\right] \tag{7}$$

Bohachevsky Function is a multimodal and separable function with ample local minimum.

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \tag{8}$$

The Bohachevsky is a bowl shaped function where a global minima $f(x^*) = 0$ is bounded to $[-100,100]$ [24].

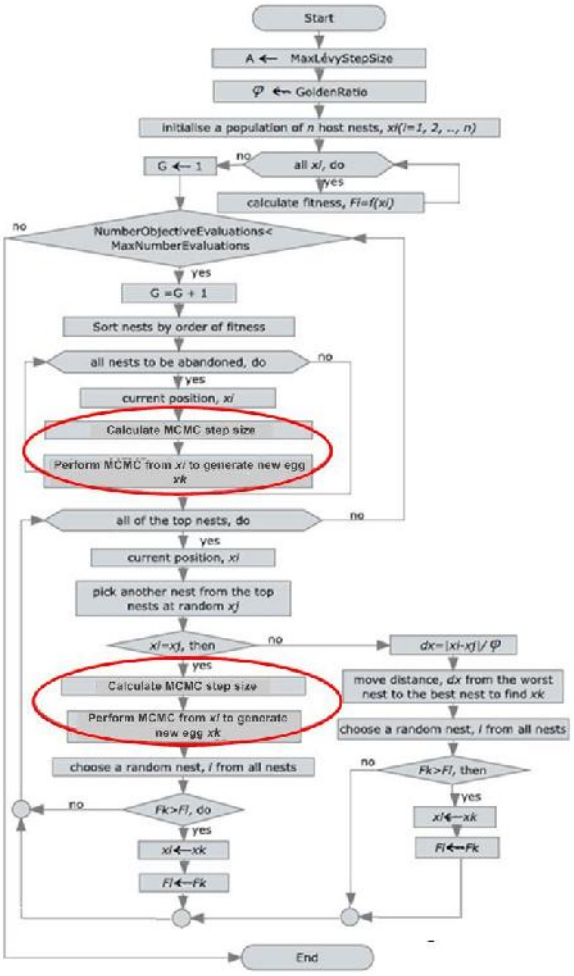

Fig. 1. MCS-MCMC flowchart

B. Results

In order to compare both MCS-Lévy and MCS-MCMC algorithms, we have accomplished substantial simulations. We run each algorithm 30 times in order to carry out meaningful analysis. During the experiments, 15 host nests with an egg survival probability of 0.25 were used. The maximum iterations for all the algorithms are set to 100 with a total of 30 simulation runs on each function. We monitor the performances by comparing the best results, average best results, standard deviation, worst results and average worst results of both algorithms. Standard deviation is used to find any variations in the average trial values. The less indicates the better. The system used for the simulations are practically instantaneous with an Intel Core™ i5 processor with 4 GB RAM. The proposed MCS-MCMC is implemented using MATLAB7.10 on Windows 7 Ultimate.

TABLE I. EXPERIMENTAL RESULTS OF BENCHMARK FUNCTIONS

Function	Dimension	Performance Metrics	MCS-Lévy	MCS-MCMC
Ackley	50	Best	0.2104	0.072753
		Average Best	0.36533	0.10945
		Standard Deviation	0.091886	0.018187
		Worst	11.7854	1.22556
		Average Worst	4.9434	0.83814
	120	Best	0.44524	0.13416
		Average Best	0.62211	0.18207
		Standard Deviation	0.10891	0.019736
		Worst	8.8806	1.2569
		Average Worst	5.5384	0.82343
Rosenbrock's	10	Best	6.821	5.7907
		Average Best	68.553	36.0144
		Standard Deviation	93.7092	46.4826
		Worst	95728937	1214132
		Average Worst	6346387	53851.24
Bohachevsky	2	Best	0.00E+00	0.00E+00
		Average Best	5.06E-07	6.24E-07
		Standard Deviation	2.58E-06	2.64E-08
		Worst	392.7252	2.597538
		Average Worst	31.8671	1.30814

As aforesaid test functions which can be seen in Table I, the proposed MCS-MCMC algorithm improve the efficiency of the original MCS-Lévy which significantly performs better. Multimodal functions can be considered as functions that have numerous numbers of local minima. These functions, in a certain degree, more difficult to optimise. For Ackley $(d = 50)$, MCS-MCMC outperformed MCS-Lévy by 25.7% while for Ackley $(d = 120)$, MCS-MCMC outperformed MCS-Lévy by 23.2%. For Rosenbrock's, MCS-MCMC outperformed MCS-Lévy by 45.9%. Rosenbrock's, which is a unimodal test function, MCS-MCMC algorithm fitness value is considerably closer to global optima, which contrast with the original MCS-Lévy algorithm. For Bohachevsky, MCS-MCMC and MCS-Lévy are on a par, however the MCS-Lévy provides worst result by 99.3% which significantly makes the MCS-MCMC more robust compared to the MCS-Lévy.

C. Convergence

The plots of convergence were compared for both algorithms in this section. We estimated the algorithm which has better potential and converge faster approaching optimality. In Fig. 2 to 5, we outlined the absolute value of the fitness value, $f(x)$. For Ackley $(d = 50, d = 120)$ and Rosenbrock's function (refer to Fig. 2 to 4), we can portray that the proposed MCS-MCMC converged quickly to its optimal solution.

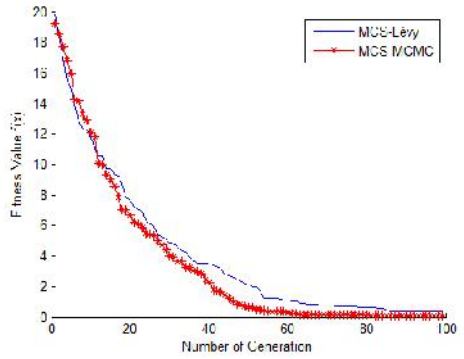

Fig. 2. The Fitness Value of Number of Generation (Ackley, *d=50*)

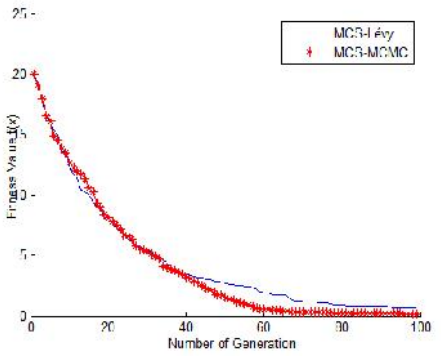

Fig. 3. The Fitness Value of Number of Generation (Ackley, *d=120*)

It can also be seen from Fig. 4, for Rosenbrock's, both algorithms converges quickly. As for MCS-MCMC, it converged when Number of Generation = 12 while MCS-Lévy when Number of Generation = 7.

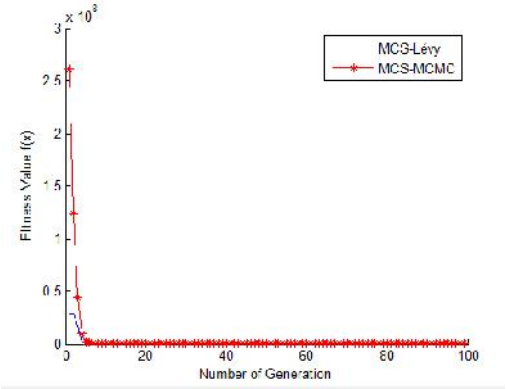

Fig. 4. The Fitness Value of Number of Generation (Rosenbrock's)

As can be seen in Fig. 5, at the first 10 number of generation, the MCS-Lévy converge quickly. However, it rises after some point. Compared to the MCS-MCMC, it reduces uniformly and converged.

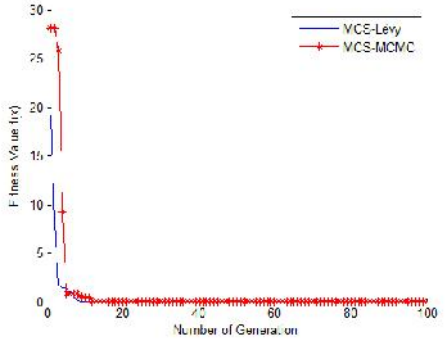

Fig. 5. The Fitness Value of Number of Generation (Bohachevsky)

V. CONCLUSION

This paper introduced the enhancement of original MCS, by replacing the Lévy flight found in the original MCS with MCMC random walk. The capability of the proposed MCS-MCMC algorithm was investigated through the performance of several experiments on well-known test functions. The results attained by the proposed MCS-MCMC algorithm are satisfying with regards to the rate of convergence. A further extension of the MCS-MCMC algorithm will be to solve multiobjective optimisation problems more naturally and more efficiently instead of focusing on the optimisation with a single objective or a few criteria with linear and nonlinear constraints.

ACKNOWLEDGMENT

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of High Education (MOHE) for financially supporting this research under Fundamental Research Grant Scheme (FRGS), Vote No 1235.

REFERENCES

[1] Sim, K. and W. Sun, *Ant colony optimization for routing and loadbalancing: survey and new directions.* IEEE T Syst Man Cy A, 2003. **33**(5): p. 560-572.

[2] Qing, A., *Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems.* IEEE T Geosci Remote, 2006. **44**(1): p. 116-125.

[3] Zhang, J., Y. Zhang, and R. Gao. *Genetic algorithms for optimal design of vehicle suspensions*. in *IEEE International Conference on Engineering of Intelligent Systems*. 2006.

[4] Sickel, J., K. Lee, and J. Heo, *Differential evolution and its applications to power plant control*, I. International Conference on Intelligent Systems Applications to Power Systems, Editor. 2007. p. 1-6.

[5] Marinakis, Y., M. Marinaki, and G. Dounias, *Particle swarm optimization for pap- smear diagnosis.* Expert Syst Appl, 2008. **35**(4): p. 1645-1656.

[6] Serrurier, M. and H. Prade, *Improving inductive logic programming by using simulated annealing.* Inform Sciences, 2008. **78**(6): p. 1423-1441.

[7] Bakar, S.Z.A., et al., *Implementation of Modified Cuckoo Search Algorithm on Functional Link Neural Network for Climate Change Prediction via Temperature and Ozone Data*, in *Recent Advances on Soft Computing and Data Mining*. 2014. p. 239-247.
[8] Yang, X.S. and S. Deb. *Cuckoo search via Levy flights*. in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09*. 2009. India: IEEE Publications.
[9] Storn, R. *Differential evolution design of an IIR-filter*. in *IEEE International Conference on Evolutionary Computation*. 1996. Nagoya.
[10] Gandomi, A., X.-S. Yang, and A. Alavi, *Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems.* Engineering with Computers, 2012.
[11] Kaveh, A. and T. Bakhshpoori, *Optimum design of steel frames using cuckoo search algorithm with Levy flights,Structural Design of Tall and Special Buildings*. 2011.
[12] Valian, E., S. Mohanna, and S. Tavakoli, *Improved cuckoo search algorithm for feed forward neural network training.* International Journal of Artificial Intelligence & Applications, 2011. **2**(3): p. 36-43.
[13] Walton, S., et al., *Modified cuckoo search: A new gradient free optimisation algorithm.* Chaos, Solitons & Fractals, 2011. **44**(9): p. 710-718.
[14] Pearson, K., *The problem of the random walk.* Nature, 1905. **72**(1865): p. 294.
[15] Bergman, C.M., J.A. Schaefer, and S. Luttich, *Caribou movement as a correlated random walk.* Oecologia, 2000. **123**(3): p. 364-374.
[16] Kilian, L. and M.P. Taylor, *Why is it so difficult to beat the random walk forecast of exchange rates?* Journal of International Economics, 2003. **60**(1): p. 85-107.
[17] Travaglione, B.C. and G.J. Milburn, *Implementing the quantum random walk.* Physical Review A, 2002. **65**(3): p. 032310.
[18] Mandelbrot, B.B., *The Fractal Geometry of Nature*. Updated and augm. ed. 1982, New York: W. H. Freeman.
[19] University, L. *Markov Chain Monte Carlo*. Available from: http://www.lancaster.ac.uk/pg/jamest/Group/stats3.html.
[20] Chattopadhyay, R., *A study of test functions for optimization algorithms.* Journal of Optimization Theory and Applications, 1971. **8**(3): p. 231-236.
[21] Schoen, F., *A wide class of test functions for global optimization.* Journal of Global Optimization, 1993. **3**(2): p. 133-137.
[22] Ackley, D.H., *An Empirical Study of Bit Vector Function Optimization.* 1987.
[23] Rosenbrock, H.H., *An Automatic Method for Finding the Greatest or Least Value of a Function.* The Computer Journal, 1960. **3**(3): p. 175-184.
[24] Hedar, A.-R. *Bohachevsky Function*. 2016; Available from: www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page595.htm.