TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Nr. 62, Vol. I, March, 2019

A BENCHMARK STUDY OF KGCS OPTIMIZATION ALGORITHM

Florina ŞERDEAN, Adina CRIŞAN

Abstract: *This paper presents the study conducted on the optimization algorithm called KGCS (Knowledge Gradient Cuckoo Search) using benchmark functions. The algorithm is highly effective due to the combination of the specific characteristics of the Cuckoo Search algorithm and the ones of the Knowledge Gradient policy. The paper also presents the benchmark mathematical functions usually used to test optimization algorithms such as Rosenbrock, Griewank or Ackley's function. The results of the conducted tests are compared to the results obtained using the unenhanced evolutionary algorithms in order to prove efficiency of the KGCS algorithm.*

Key words: *evolutionary algorithm, optimization, benchmark functions, Cuckoo Search algorithm.*

1. INTRODUCTION

Problems that require an optimal solution can be found in all fields, including engineering. For such problems one wants to identify the best solution that meet certain criteria and some given constraints.

A great part of the research has been dedicated to studying optimization methods especially for solving engineering problems [1-5] and to testing them. Due to the constantly changing market there is an even greater need now than before of optimization algorithms characterized by high performance. This can be quantified by how fast it finds the optimum solutions of the mathematical benchmark functions that exemplify a series of possible categories of real-world applications.

There are several published scientific papers dedicated to the testing of optimization algorithms using benchmark functions. For example, in [6] the authors evaluate the performance of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms regarding their general applicability as numerical optimization techniques. The comparison is performed on a suite of several widely used benchmark problems, as well as on two noisy functions. Other papers such as [7] present the testing of several algorithms on mathematical benchmark functions algorithms, but also apply the best one to a calibration problem for a water distribution system. In general, all tests for optimization algorithms are performed on the same batch of benchmark functions that can be found in [8] and [9], where the functions were divided into three categories: unimodal functions, functions with many local minima, and functions with a few local minima.

2. ALGORITHM DESCRIPTION

2.1 Cuckoo Search algorithm

The category of meta-heuristics that were created based on the evolution and/or functioning of different biological systems from the real world have shown great potential for solving optimization problems. Cuckoo Search (CS) is one of these meta-heuristics and is inspired by the specific habits that cuckoos have regarding egg laying and breeding.

CS algorithm has been tested and used for in many domains including engineering design, a domain of great industrial interest [10]. One of the variations of this algorithm uses Lèvy flights to mimic the flight of the cuckoos and this variation of the algorithm was implemented and tested for the research presented in this paper. The implemented optimization algorithm uses a random initial population. Each cuckoo lay one egg in a host bird nest. The host nests are chosen usually in the proximity of the cuckoo's current location by using Lèvy flights. The flights obviously take into consideration the search space limits. Fig.1 shows an example of A 3D Lèvy flight of 500 steps with (0; 0; 0) as starting point. When analyzing the flight, it can be easily observed that the flight has an aggregation of small steps in the proximity of the origin point but also a few extreme jumps. This aspect ensures a thorough local search while not limiting the search to just one area. The next steps consist in evaluating each new cuckoo chick that hatches from the laid eggs and updating the value of the best cuckoo.

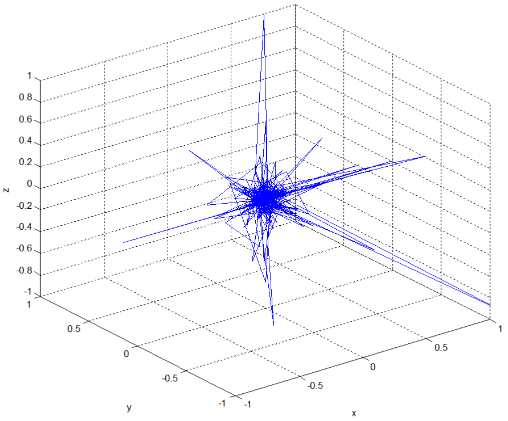


Fig. 1. A 3D Lèvy flight of 500 steps with (0,0,0) as starting point [11].

When studying the life of real cuckoo it can be seen that there are cases when the eggs are discovered by the host bird. Hence, a constant representing the probability to discover the cuckoo eggs is used. Based on this probability some of the laid eggs are discovered and removed from the nest and new eggs are laid. Proceeding like this the cuckoo population is not allowed to migrate too fast towards an area where the optimal solution seems to be. In this way, the chances of finding a local optimum in optimization problems is reduced. At the end of each generation, the cuckoo population is renewed by keeping the best cuckoos of the cuckoo chicks and parents.

The end condition of the cuckoo's migration towards the optimum is given by a maximum number of generations. It can also be customized by using the number of objective function evaluations. The CS algorithm was implemented in Matlab and was tested on stochastic benchmark functions [11].

2.2 KGCS algorithm

In general, all complex optimization problems need thousands or hundreds of thousands of objective function evaluations and therefore, lead to high computational costs. A lot of research has been focused into developing robust and efficient heuristics. The already presented CS algorithm was improved based on the concept that with more knowledge, the length of the exploration phase can be diminished. More details regarding how the growth in knowledge of the cuckoo population during the generational process was evaluated using the Knowledge Gradient policy are presented in [11, 12].

The improved KGCS algorithm has two main phases. The first phase corresponds to the exploration step. Basically, the algorithm starts with three initially random cuckoo populations that explore the search space separately. When a new generation of cuckoos is obtained, the best cuckoo, as well as the archive of best cuckoos is updated for each of the three populations. This first phase finishes after a few generations (approx. 5%-10% from the maximum allowed number of generations).

The second phase, even if it has a certain level of exploration, is focused on exploitation. First, the knowledge gradient is computed for each of the three cuckoo populations based on their own archive. Then, only the population with the largest expected enhancement according to Knowledge Gradient policy will remain in the second phase and will continue the search. The above presented enhanced heuristic was implemented in Matlab since vectorization is one of its core concepts and therefore, it is computationally inexpensive in terms of both memory requirements and speed.

3. BENCHMARK STUDY

3.1 Introduction

Real-world optimization problems involve complex objective functions whose graphical representation is often characterized by several peaks or valleys where local optimum values can be found or by only one hard-to-find optimum. In order to validate a certain metaheuristic and to be able to draw some conclusions about its performance, a test function set is recommended to be used.

Two of the chosen multimodal functions, Rastrigin and Griewank's functions, are usually used for testing evolutionary algorithms (for more details see [13 - 15]).

3.2 Rastrigin’s function

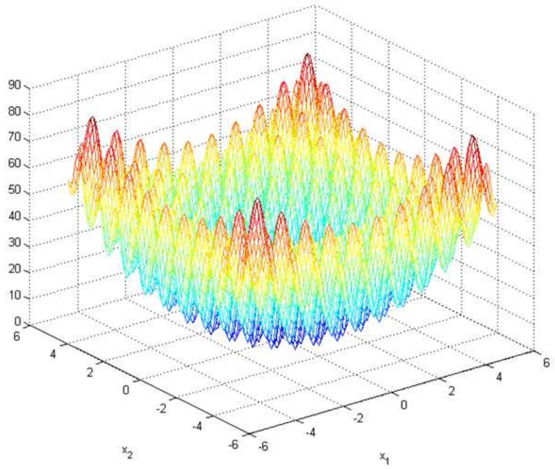


Fig. 2. The plot for Rastrigin's 2D function [11]

Rastrigin's function has proven to be quite a challenge for genetic algorithms due to the large search space and large number of local minima whose value increases as the distance to the global minimum increases [13]. The function is given by:

$$f : \mathbb{R}^n \to \mathbb{R},\ f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos\left(2\pi x_i\right)\right], \quad (1)$$

$n \in \mathbb{N}^*$

but the test area is usually narrowed to -5.12 ≤ x_i ≤ 5.12, for i = 1, 2, …, n. The global minimum is obtained for x_i = 0, for i = 1, 2, …, n and the plot of Rastrigin's 2D function is illustrated in Fig. 2.

3.3 Griewank's function

Griewank's function has a product term that introduces interdependence among the variables. The aim is the failure of the techniques that optimize each variable independently [13]. According to [15], this function is the only scalable, nonlinear and nonseparable function from the common test functions. The function is given by:

$$f : \mathbb{R}^n \to \mathbb{R},\ f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (2)$$

$n \in \mathbb{N}^*$

but the test area is usually narrowed to -600 ≤ x_i ≤ 600, for i = 1, 2, …, n. The global minimum is obtained for x_i = 0, for i = 1, 2, …, n and the plot of Griewank's 2D function is illustrated in Fig. 3.

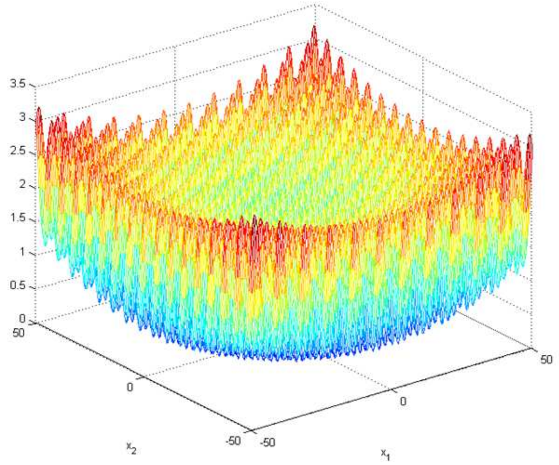


Fig. 3. The plot for Griewank's 2D function [11]

3.4 Ackley’s function

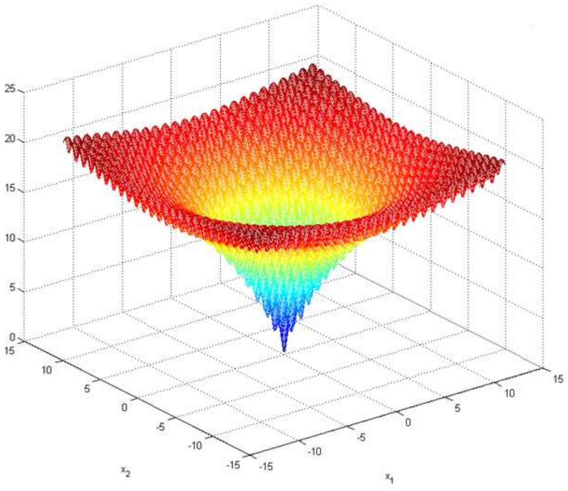


Fig. 4. The plot for Ackley's 2D function [11]

The third multimodal function chosen for testing the algorithm was Ackley's function. The main reason for this choice was the fact that the tested search strategy is required to use an efficient combination of exploratory and exploitative components in order to obtain good results for this function, which is given by:

$$f:¡^{n}\to¡,\ f(x)=-a\cdot\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)- \qquad (3)$$

$$-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(c\cdot x_i\right)\right)+a+\exp(1),\ n\in¥^{*}$$

The following values are recommended to be used: a = 20, b = 0.2, c = 2π and the test area is usually narrowed to $-32.768 \le x_i \le 32.768$, for *i* = 1, 2, ..., *n*. The global minimum is obtained for $x_i = 0$, for *i* = 1, 2, ..., *n* and the plot of Ackley's 2D function is illustrated in Fig. 4.

3.5 Rosenbrock's function

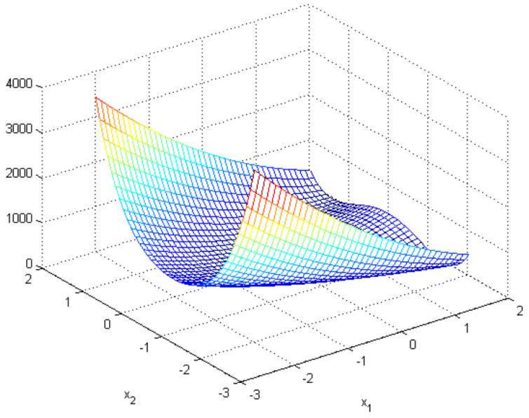


Fig. 5. The plot for Rosenbrock's 2D function [11]

Another function used for testing the implemented algorithm was Rosenbrock's function. The global minimum is situated in a deep valley with the shape of a parabola. Finding the valley is easy, but, due to the non-linearity of the valley, many algorithms converge slowly towards the optimum. It is considered by many authors as a challenge for any optimization algorithm because of the non-linear interaction between its variables. The function is given by:

$$f:¡^{n}\to¡,\ f(x)=\sum_{i=1}^{n}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2\right],\ (3)$$

$$n\in¥^{*}$$

The test area is usually narrowed to $-2.048 \le x_i \le 2.048$ for *i* = 1, 2, ..., *n*. The global minimum is obtained for $x_i = 1$, for *i* = 1, 2, ..., *n* and the plot of Rosenbrock's 2D function is illustrated in Figure 5.

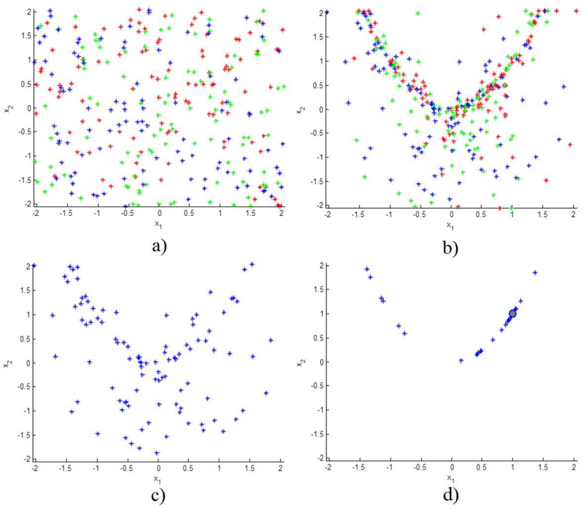
4. TESTING

First, the implemented CS algorithm was tested 10 times on each of the mentioned benchmark functions. The results are summarized in Table 1. As it can be observed in the mentioned table, for each test function the algorithm has found the global minimum every time, requiring less than 206,000 evaluations of the objective function on average.

Table 1. Test results for CS algorithm [11]

Function	Average number of required objective function evaluations	Percentage of identifying the solution
Rastrigin	205,178.6	100%
Griewank	199,214.3	100%
Ackley	177,571.4	100%
Rosenbrock	195,178.6	100%

The improved KGCS algorithm was also tested on some benchmark functions such as Rosenbrock, Griewank, and Ackley's function. The evolution of the populations was monitored during the two phases of the algorithm for each of the tested functions. An intense exploration of the search space was observed during the first phase as it can be seen in the Fig. 6 - 8, where the populations are represented with three different colors: red for the first population; green for the second population; blue for the third population.



a)

b)

c)

d)

Fig. 6. Test conducted on Rosenbrock's 2D function: a) The beginning of the first phase; b) The end of the first phase; c) The beginning of the second phase; d) The end of the second phase [11]

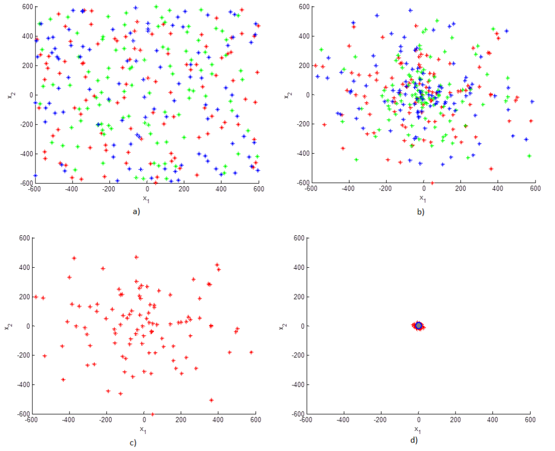


Fig. 7. Test conducted on Griewank's 2D function: a) The beginning of the first phase; b) The end of the first phase; c) The beginning of the second phase; d) The end of the second phase [11]

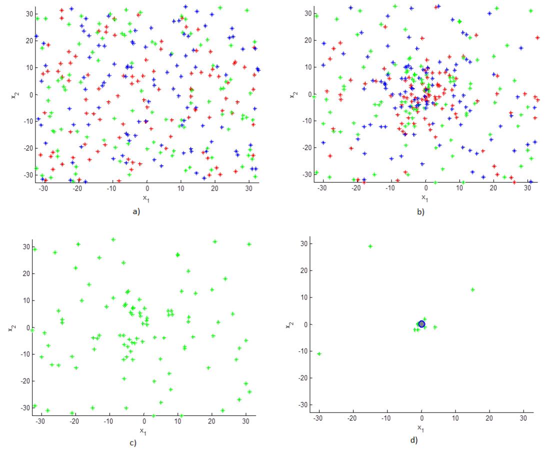


Fig. 8. Test conducted on Ackley's 2D function: a) The beginning of the first phase; b) The end of the first phase; c) The beginning of the second phase; d) The end of the second phase [11]

Table 2. Test results for KGCS algorithm [11]

Function	Average number of required objective function evaluations	Percentage of identifying the solution
Rosenbrock	196,621.4	100%
Griewank	186,350.5	100%
Ackley	171,978.6	100%

During the second phase of the algorithm the population that according to KG policy will bring the most significant improvement in the remaining generations is searching for the best solution. As it can be seen in Fig. 6-8 the second phase is characterized by an increasing level of convergence.

The CS algorithm improved by using KG policy was tested 10 times on each of the mentioned benchmark functions. The results are summarized in Table 2.

A comparison is required between the results obtained using KGCS algorithm and the ones obtained using the standard version of CS algorithm. As it can be seen in Table 1 and Table 2, both algorithms have a percentage of identifying the global optimum of 100%. However, there is a decrease of the number of evaluations required to find the global optimum when using KGCS. The average decrease of the number of required objective function evaluations caused by using KGCS is about 12,005 evaluations, which means a percentage decrease of 6.13 % from the number of required objective function evaluations when using standard CS.

Even if the improvement of 6.13% seems to be not so important, it can become crucial when it comes to complex optimization functions which require a long running time. Moreover, it is very likely that more runs are required to obtain an accurate solution when the process is governed by randomness and in this case any decrease of the running time is a desirable benefit. Also, the percentage decrease might differ when KGCS is used for other functions. For example, for the optimal design problem solved in [12], the percentage decrease was 13% for the first stage and approximately 17%, for the second and third stage.

5. CONCLUSIONS

The standard CS algorithm was improved by incorporating the Knowledge Gradient policy for evaluating and predicting the knowledge of the cuckoo populations. The current paper briefly presents the standard as well as the enhanced CS algorithm.

Both algorithms were tested on a set of mathematical benchmark functions. The obtained results showed that KGCS is faster than the standard CS algorithm and it is suitable to be used for solving multimodal optimization problems.

6. REFERENCES

[1] Parkinson, A. R., Balling, R. J., Hedengren, J. D., *Optimization methods for engineering*

design. Applications and Theory, Brigham Young University, 2013

[2] Ovidiu Buiga, Simion Haragâş, Optimal design with evolutionary algorithms of a gear coupling, Acta Technica Napocensis - Series: Applied Mathematics and Mechanics, Vol 54, No 2, 2011, pp. 273-276

[3] Ovidiu Buiga, Simion Haragâş, A 2 stage coaxial helical speed reducer gearings optimal design with genetic algorithms, Acta Technica Napocensis - Series: Applied Mathematics and Mechanics, Vol 55, No 3, 2012, pp. 535-542

[4] Iuliu Negrean, Formulations on input parameters in advanced dynamics, Acta Technica Napocensis - Series: Applied Mathematics and Mechanics, Vol 61, No 3, 2018, pp. 305-312

[5] Iuliu Negrean, Advanced Notions in Analytical Dynamics of Systems Acta Technica Napocensis - Series: Applied Mathematics and Mechanics, Vol 60, No 4, 2017, pp. 491-502

[6] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 2004, pp. 1980-1987 Vol.2. doi: 10.1109/CEC.2004.1331139

[7] Jung BS, Karnev BW Lambert MF (2006) Benchmark tests of evolutionary algorithms: mathematic evaluation and application to water distribution systems. J Env Inf 7(1):24–35

[8] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” IEEE Trans. Evol. Comput., vol. 3, no. 2, p. 82, Jul. 1999.

[9] C. Y. Lee and X. Yao, “Evolutionary programming using mutations based on the Lévy probability distribution,” IEEE Trans. Evol. Comput., vol. 8, no. 1, pp. 1–13, Feb. 2004

[10] Yang, X.-S., and Deb, S., *Engineering Optimisation by Cuckoo Search*, Int. J. Mathematical Modelling and Numerical Optimisation, Vol. 1, No. 4, 330–343, 2010

[11] Rusu, M.-F., Optimizations under uncertainty with applications in rolling bearing industry, Faculty of Machine Building, Technical University of Cluj-Napoca, 2014 (PhD. Thesis)

[12] Tudose, L., Rusu, F., Tudose, C., Optimal design under uncertainty of bearing arrangements, Mechanism and Machine Theory, 98, 2016, pp. 164-179

[13] Gordon, V. S., and Whitley, L. D. Serial and parallel genetic algorithms as function optimizers. In Proceedings of the 5th International Conference on Genetic Algorithms (San Francisco, CA, USA, 1993), Morgan Kaufmann Publishers Inc., pp. 177-183.

[14] Jamil, M., and Yang, X.-S. A literature survey of benchmark functions for global optimization problems. Int. Journal of Mathematical Modelling and Numerical Optimisation 4, 2 (2013), 150-194.

[15] Whitley, D., Mathias, K., Rana, S., and Dzubera, J. Building better test functions. In Proceedings of the Sixth International Conference on Genetic Algorithms (1995), Morgan Kaufmann, pp. 239-246.

UN STUDIU DE REFERINȚĂ ASUPRA ALGORITMULUI DE OPTIMIZARE KGCS

Această lucrare prezintă studiul realizat asupra algoritmului de optimizare numit KGCS folosind funcții de test standard. Algoritmul este foarte eficient datorită îmbinării caracteristicilor algoritmului Cuckoo Search cu cele ale metodei Knowledge Gradient (KG). Lucrarea prezintă de asemenea funcțiile matematice folosite în general pentru a testa algoritmii de optimizare, funcții precum Rosenbrock, Griewank sau Ackley. Rezultatele testelor au fost comparate cu rezultatele obținute folosind algoritmul evolutiv neîmbunătățit cu scopul de a dovedi eficiența algoritmului KGCS.

Florina ȘERDEAN, PhD. Math., Lecturer, Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, Florina.Rusu@omt.utcluj.ro.

Adina CRIȘAN, PhD. Eng., Lecturer, Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, ducaadina@yahoo.com.