

## Research Article

# Integrated Optimization of Differential Evolution with Grasshopper Optimization Algorithm

Duangjai Jitkongchuen\*, Udomlux Ampant

*College of Innovative Technology and Engineering, Dhurakij Pundit University, Thailand***ARTICLE INFO***Article History*

Received 24 March 2018

Accepted 6 June 2018

*Keywords*Meta-heuristic  
differential evolution algorithm  
grasshopper optimization algorithm  
optimization**ABSTRACT**

This paper proposes a scheme to improve the differential evolution (DE) algorithm performance with integrated the grasshopper optimization algorithm (GOA). The grasshopper optimization algorithm mimics the behavior of grasshopper. The characteristic of grasshoppers is slow movement in the larval stage but sudden movement in the adulthood which seem as exploration and exploitation. The grasshopper optimization algorithm concept is added to DE to guide the search process for potential solutions. The efficiency of the DE/GOA is validated by testing on unimodal and multimodal benchmarks optimization problems. The results prove that the DE/GOA algorithm is competitive compared to the other meta-heuristic algorithms.

© 2018 The Authors. Published by Atlantis Press SARL.

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

## 1. INTRODUCTION

Meta-heuristic algorithm techniques constitute range from simple local search procedures to complex learning processes. The basic concepts of meta-heuristics permit an abstract level description. Nature-inspired meta-heuristic can be grouped in three main classes: Evolution-based, physics-based and swarm-based.

Evolution-based algorithms are inspired by the concepts of nature evolution. The population in search process are randomly generated which are developed over subsequent generation. The next generation is created by combination of the individuals in the previous generation. These provide the population to be optimized over the course of generations. The most popular evolution-based algorithm is genetic algorithms (GA) [1]. GA has a process of fitness-based selection and recombination to produce a successor population for the next generation. During recombination, child chromosomes are produced by recombined genetic material from parent chromosomes are selected. As this process is repeated, a sequence of consecutive generations evolves and the average fitness of the chromosomes inclines to increase until some stopping criterion is reached. Other popular algorithms such as differential evolution (DE) algorithms [2], the principal difference between genetic algorithms and differential evolution is that DE use mutation as the primary search mechanism, while GA use crossover as the probabilistic mechanism to exchange of information among solutions to locate better solutions. Biogeography-based optimizer (BBO) [3] algorithm is applied from nature phenomenon regarding the distribution of living creatures in various islands. In this field of study different ecosystems are explored for finding the relations

between different species in terms of immigration, emigration, and mutation. The main inspiration of the BBO algorithm is to reach a stable situation based on migration and mutation in the evolution of ecosystems.

The second class of meta-heuristic is physics-based algorithms that imitate the physical rules. For example, gravitational search algorithm (GSA) [4], GSA is a nature-inspired conceptual framework based on gravitational kinematics, a branch of physics that models the motion of masses moving under the influence of gravity. Galaxy-based search algorithm [5] is inspired on the spiral arm of spiral galaxies to search its surrounding. This spiral movement is improved by chaos to escape from local optimums. Curved space optimization [6] inspired by general relativity theory is used to enhance the efficiency of a simple random search, and convert it to a very robust optimization tool.

The third class of meta-heuristic is swarm-based algorithms that imitate the social behavior of swarms, herds, flocks, or schools of animals. The most popular algorithms are particle swarm optimization (PSO) [7]. PSO inspired by bird flocking or fish schooling. Each particle will be iteratively updated to improve its solution during movement around the search space. The result leads to the best value of global solution. Other popular swarm-based algorithms are bat algorithms (BA) [8]. BA is inspired by echolocation behavior of microbats to avoid obstacles, find their home, and detect their prey in the darkness. Firefly algorithm (FA) [9] applied from the flashing light of fireflies when they want to mate or to warn others about predators.

The new meta-heuristic has been focused on this paper is differential evolution algorithm which is a global search optimization algorithm and improved performance of it by added grasshopper optimization algorithm (GOA) to increase convergence rate.

\*Corresponding author. Email: [duangjai.jit@dpu.ac.th](mailto:duangjai.jit@dpu.ac.th)

The rest of the paper is structured as follows. Section 2 describes some backgrounds of the differential evolution algorithm. In Section 3, the proposed algorithm is presented. Section 4 shows the experimental results and the conclusions will be discussed in Section 5.

## 2. THE DIFFERENTIAL EVOLUTION ALGORITHM

The differential evolution algorithm optimizes a problem by sustaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple procedure, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand.

The procedure of differential evolution algorithm starts to initialize target vectors  $X_i^G = (x_{1,i}, x_{2,i}, \dots, x_{j,i}, \dots, x_{N,i})$  where  $i = 1, 2, \dots, NP$ . NP is the population size.  $N$  is the dimension of the population. The superscript  $G$  identifies the  $G^{th}$  generation.

The target vectors are used to generate the mutant vectors  $V_i^G = (v_{1,i}, v_{2,i}, \dots, v_{j,i}, \dots, v_{N,i})$  in next steps. The DE/current-to-best/1 mutation scheme is shown in Eq. (1).

$$V_i^G = X_i^G + F(X_{best}^G - X_i^G) + F(X_{r_1}^G - X_{r_2}^G) \quad (1)$$

where  $r_1$  and  $r_2$  are integer number chosen from the set  $\{1, 2, \dots, NP\}$  and must be different from index  $i$ .  $X_{best}^G$  is the best solution in the generation  $G$ .  $F$  is used to control amplification of the differential evolution. The value of  $F$  is set in the range  $[0, 2]$ . The target vectors and mutant vectors are used to generate the trial vectors  $U_i^G = (u_{1,i}, u_{2,i}, \dots, u_{j,i}, \dots, u_{N,i})$  in crossover operation to increase the diversity of population. The trial vectors are generated by Eq. (2).

$$u_{j,i} = \begin{cases} v_{j,i}, & \text{if (rand} \leq \text{CR) or } (j = j_{rand}) \\ x_{j,i}, & \text{otherwise} \end{cases} \quad (2)$$

where  $j = 1, 2, \dots, N$ . rand is a uniform random number chosen in the range  $[0, 1]$ . CR  $\in [0, 1]$  is the crossover parameter.  $j_{rand}$  is an index randomly chosen in the range  $[0, N]$ .

The last step is the selection operation to choose a better vector for next generation. The new generations are selected by comparing the fitness value between the target vector and the trial vector. The selection is shown in Eq. (3).

$$X_i^{G+1} = \begin{cases} U_i^G, & \text{if } (f(U_i^G) < f(X_i^G)) \\ X_i^G, & \text{otherwise} \end{cases} \quad (3)$$

The whole process is repeated until the termination criteria are satisfied or a predefined number of iterations are reached.

## 3. THE PROPOSED ALGORITHM

The schema of DE/current-to-best/1 in mutation operation is updated by splitting  $F$  mutation parameter to  $\lambda$  and  $F$ . The new mutation schema is shown in Eq. (4).

$$V_i^G = X_i^G + \lambda(X_{best}^G - X_i^G) + F(X_{r_1}^G - X_{r_2}^G) \quad (4)$$

where  $\lambda$  is used to balance the difference vectors between the best vectors and the target vectors. The value of  $\lambda$  that is generated by Eq. (5) is decreased in each generation.

$$\lambda^{G+1} = \lambda^G - (\lambda^G \times c_1) \quad (5)$$

Meanwhile the mutation parameter  $F$  is used to perform the amplification of the difference between two random population members. When the procedure is trapped in the local optimum, the value of  $F$  is increased by Eq. (6).

$$F^{G+1} = F^G + (F^G \times c_2) \quad (6)$$

The parameters  $c_1$  in Eq. (5) and  $c_2$  in Eq. (6) are uniform random numbers.  $c_1 \in [0, 0.1]$  is a random value and  $c_2 \in [0, 1]$  is a random value. The crossover parameter (CR) is regenerated at the end of each generation to find an optimized parameter. The new crossover parameter is generated by Eq. (7).

$$CR^{G+1} = CR^G + (\text{mean}_A(S_{CR}) \times c_3) \quad (7)$$

where  $\text{mean}_A$  is the usual arithmetic mean.  $S_{CR}$  is the set of crossover parameter values.  $C_3 \in [0, 1]$  is a uniform random number.

After that, the crossover operation is rearranged. The vectors that created from the target vectors and the mutant vector is called the prime vectors ( $U_i^G$ ). The prime vectors is used to generate the trial vectors by comparing the fitness value between the prime vectors and the mutant vectors, otherwise the trial vectors is generated by grasshopper optimization algorithm.

The grasshopper optimization algorithm was proposed in Saremi et al. [10] mimics the behavior of grasshopper. The characteristic of the swarm in the larval stage is slow movement and small steps but is long range and suddenly movement in adulthood. These two characters seem as exploration and exploitation. The search agents are encouraged to move suddenly in exploration and conduce to move locally during exploitation. These two functions as well as target seeking are performed by grasshopper naturally.

The proposed algorithm takes the feature of movement of the grasshopper to generate a new value for the trial vectors is showed in Eq. (8).

$$U_i^d = c \left( \sum_{\substack{j=1 \\ j \neq i}}^{NP} c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{i,j}} \right) + \hat{T}_d \quad (8)$$

where  $ub_d$  and  $lb_d$  are the upper bound and the lower bound in the  $d^{th}$  dimension.  $s(r) = fe^{-r/l} - e^{-r}$ ,  $d_{i,j}$  is the distance between  $i^{th}$  and  $j^{th}$  grasshopper.  $\hat{T}_d$  is the value of the  $d^{th}$  dimension in the target. The parameter  $c$  is decreased equivalent to the number of iteration to balancing exploration and exploitation is calculated as Eq. (9).

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L} \quad (9)$$

where  $c_{max}$  and  $c_{min}$  are the maximum and the minimum value, respectively.  $l$  is the current iteration and  $L$  is the maximum number of iterations.

### 4. THE EXPERIMENTAL RESULTS

The proposed algorithm has been evaluated performance with thirteen benchmark functions [11]. The test functions are unimodal ( $f_1-f_7$ ) and multimodal ( $f_8-f_{13}$ ). The details of benchmark functions are shown in Table 1. The mutation parameters set  $\lambda$ ,  $F = 0.5$ , the crossover parameter set  $CR = 0.9$  and population size set  $NP = 100$ .

The proposed algorithm was run 30 times on each benchmark functions. The average results are summarized in Table 2. The performance of the proposed algorithm is compared with two EAs: DE and fast evolutionary programming (FEP) and compared with swarm-based algorithm: PSO. The results of DE, FEP and PSO were taken from the results reported in Saremi et al. [12].

According to the results of Table 2, DE/GOA is able to provide very competitive results. This algorithm outperforms all others in  $f_1, f_2, f_3$ , and  $f_7$ . Both DE/GOA and DE were successful to find the optimal solution in  $f_4$  and  $f_6$ . In the function  $f_5$ , only DE could solve the optimum. Therefore, these results show the performance of DE/GOA in terms of exploiting the optimum.

The multimodal functions have many local optima with the number increasing exponentially with dimension. This makes them suitable for benchmarking the exploration ability of an algorithm.

The experimental showed that DE/GOA is able to provide very competitive results on the multimodal benchmark functions as well.

The experimental results show that the global search algorithms which added grasshopper optimization algorithm could

Table 2 | The experimental results

Functions	DE/GOA	DE	FEP	PSO
$f_1$	2.35E-220	8.20E-14	5.70E-04	1.36E-04
$f_2$	4.28E-11	1.50E-09	8.10E-03	4.21E-02
$f_3$	3.50E-226	6.80E-11	1.60E-02	70.12562
$f_4$	0	0	3.00E-01	1.086481
$f_5$	28.98017	0	5.06	96.71832
$f_6$	0	0	0	1.02E-04
$f_7$	2.11E-05	4.63E-03	1.42E-01	1.23E-01
$f_8$	-8831.53	-11080.1	-12554.5	-4841.29
$f_9$	6.76E-10	69.20	4.60E-02	46.70423
$f_{10}$	4.44E-16	9.70E-08	1.80E-02	2.76E-02
$f_{11}$	0	0	1.60E-02	9.22E-03
$f_{12}$	2.87E-09	7.90E-15	9.20E-06	6.92E-03
$f_{13}$	3.01E-08	5.10E-14	1.60E-04	6.68E-03

Table 1 | The benchmark functions

Functions	Range
$f_1(x) = \sum_{i=1}^N x_i^2$	[-100, 100]
$f_2(x) = \sum_{i=1}^N  x_i  + \prod_{i=1}^N  x_i $	[-100, 100]
$f_3(x) = \sum_{i=1}^N (\sum_{j=1}^i x_j)^2$	[-10, 10]
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq N\}$	[-100, 100]
$f_5(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-100, 100]
$f_6(x) = \sum_{i=1}^N (\lfloor x_i + 0.5 \rfloor)^2$	[-30, 30]
$f_7(x) = \sum_{i=1}^N ix_i^4 + \text{random}[0, 1]$	[-100, 100]
$f_8(x) = \sum_{i=1}^N -x_i \sin(\sqrt{ x_i })$	[-1.28, 1.28]
$f_9(x) = \sum_{i=1}^N [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-500, 500]
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{N} \sum_{i=1}^N \cos 2\pi x_i\right) + 20 + e$	[-5.12, 5.12]
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-32, 32]
$f_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	[-600, 600]
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases} \quad y_i = 1 + \frac{x_i + 1}{4}$	[-50, 50]
$f_{13}(x) = 0.1 \{\sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[-50, 50]

increase convergence rate because the characteristic of GOA is both exploration and exploitation. Meanwhile the results of multimodal functions are not perfect that may be due to the multimodal functions have many increasing number of local optimum.

## 5. CONCLUSION

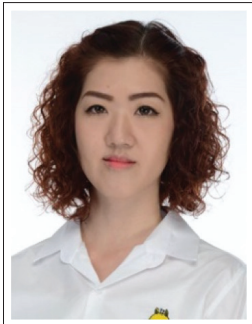
This work proposed to improve the DE algorithm performance with integrated the GOA. Thirteen benchmark functions were employed the performance in terms of exploitation and exploration. The results showed that DE/GOA was able to provide competitive results compared to well known heuristics such as DE, FEP and PSO.

## REFERENCES

- [1] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1992), 66–72.
- [2] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (1997), 341–359.
- [3] D. Simon, Biogeography-based optimization, *IEEE Transactions on Evolutionary Computation*, IEEE, 2008, pp. 702–713.
- [4] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inform. Sci.* 179 (2009), 2232–2248.
- [5] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization, *Int. J. Comput. Sci. Eng.* 6 (2011), 132–140.
- [6] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: a random search based on general relativity theory, 2012, arXiv:1208.2214 (<https://arxiv.org/abs/1208.2214>).
- [7] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, Perth, WA, Australia, 1995, pp. 1942–1948.
- [8] X.S. Yang, A New Metaheuristic Bat-inspired Algorithm, Springer, Berlin, 2010, pp. 65–74.
- [9] I. Fister, I. Fister Jr, X-S. Yang, J. Brest, A comprehensive review of firefly algorithms, *Swarm Evolution. Comput.* 13 (2013), 34–46.
- [10] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimization algorithm: theory and application, *Adv. Eng. Software* 105 (2017), 30–47.
- [11] X. Yao, Y. Liu, G. Lin, Evolutionary Programming Made Faster, *IEEE Transactions on Evolutionary Computation*, IEEE, 1999, pp. 82–102.
- [12] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software* 69 (2014), 46–61.

## Authors Introduction

### Dr. Duangjai Jitkongchuen



She graduated Ph.D. in Information Technology from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 2014 and now she is working as Assistant Professor at Dhurakij Pundit University, College of Innovative Technology and Engineering as a deputy dean administration. Her research area is about machine learning, data mining and bio-inspired algorithms.

### Ms. Udomlux Ampant



She graduated Master of Business Administration (Marketing) Dhurakij Pundit University. She is an lecturer in the College of Innovative Technology and Engineering in Dhurakij Pundit University.