10.1109/IranianCEE.2019.8786601

27th Iranian Conference on Electrical Engineering (ICEE2019)

DTO: Donkey Theorem Optimization

Mohammad Dehghani
PhD. Student
Dept. of Electrical and Electronics Engineering
Shiraz University of Technology
Shiraz, IRAN
m.dehghani@sutech.ac.ir

Mohammad Mardaneh
Associate Prof.
Dept. of Electrical and Electronics Engineering
Shiraz University of Technology
Shiraz, IRAN
mardaneh@sutech.ac.ir

Om Parkash Malik
Prof.
Dept. of Electrical Engineering
University of Calgary
Calgary Alberta Canada
maliko@ucalgary.ca

S. Mansour NouraeiPour
M.Sc.
Kogilooye and Boyerahmad Electric Energy Distribution Co.
mansour517@gmail.com

Abstract— **Metaheuristic optimization algorithms have been used in many applications in recent years. Most of these algorithms are inspired by physical processes or living beings' behaviors. A new optimization algorithm, called Donkey Theorem Optimization (DTO), that simulates the behavior of Donkeys is proposed in this paper. DTO is based on donkey theorem that mimics behavior of donkey for reach to food. Proposed algorithm is tested on 23 well-known benchmark test functions and its performance compared with eight optimization algorithms. The results show that DTO is able to provide better results as compared to the other well-known optimization algorithms.**

Keywords- Donkey; Donkey Theorem, optimization, Donkey Theorem Optimization.

I. INTRODUCTION

Various algorithms have been proposed to solve a variety of engineering optimization problems recently [1-4]. These optimization problems are very complex, because they have more than one local optimal solution. In order to increase the accuracy and efficiency of these problems, researchers have used meta-heuristic optimization algorithms [5]. Meta-heuristic algorithms are classified into Evolutionary based, Swarm based and Physics based methods.

A. Evolutionary based metaheuristic algorithms

Evolutionary based algorithms are inspired by theory of biological evolution and natural selection. Genetic Algorithm (GA), as an evolutionary algorithm [6], is based on reproduction process. Differential Evolution (DE), another evolutionary based algorithm [7], is presented to overcome the main defect of the GA, namely the lack of local search. Some of the other popular evolutionary based algorithms are Evolution Strategy (ES) [8], Biogeography based Optimizer (BBO) [9] and Genetic Programming (GP) [10].

B. Swarm based metaheuristic algorithms

These algorithms are based on the swarm action of social creatures. Particle Swarm Optimization (PSO) is modeled on social behavior of bird categories [11]. Artificial Bee Colony (ABC) is an optimization strategy that simulates the behavior of a bee colony [12]. The other swarm based meta-heuristic algorithms are Emperor Penguin Optimizer (EPO) [13], Gray Wolf Optimization (GWO) [14] and Grasshopper Optimization Algorithm (GOA) [15].

C. Physics based metaheuristic algorithms

Physics based algorithms employ the rules of physics for transfer of information between search agents. Spring Search Algorithm (SSA) is developed based on Hook's law [16]. Gravitational Search Algorithm (GSA) is an optimization method inspired by the law of gravitation between planets in space and the galaxies [17]. Some of the other such algorithms are Simulated Annealing (SA) [18], Central Force Optimization (CFO) [19] and ChParged System Search (CSS) [20].

A new optimization algorithm, named Donkey Theorem Optimization (DTO) to solve the optimization problems, is presented in this paper. DTO is inspired by the behavior of donkey to reach food by the shortest path. Performance of DTO has been evaluated on 23 benchmark test functions.

The rest of this paper is structured as follows: Section II presents the Donkey Theorem. The proposed DTO is discussed in Section III. The results are presented in Section IV. Finally, the conclusions are given in Section V.

II. DONKEY THEOREM

The Donkey Theorem is also known as the triangle inequality theorem. It states that in a triangle ABC: AC < AB + BC.

The name comes from the idea that if a donkey standing at point A, and a hay stack at point C, the donkey will always take the shorter path to go straight from A to C instead of from A to B to C. This theory is shown in Fig. 1.

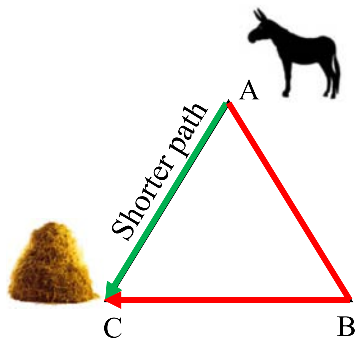

Figure 1. Donkey Theorem

III. DONKEY THEOREM OPTIMIZATION

In this paper, an optimization algorithm called Donkey Theorem Optimization, that simulates the behavior of a donkey, is designed. The DTO is described in two steps:

- Formation of an artificial system with discrete time in the problem space, initial positioning for searcher agent, setting rules and parameters.
- Repeat steps of the algorithm, until the stopping condition is satisfied.

A. Formation of the system, laws and parameters setting

In the first step, the system space is determined. The environment consists of a multidimensional coordinate system in the space of the problem definition Searching agents are donkeys looking for food. The position of each donkey is a point in space that is the answer to the problem.

Now imagine the system as a set of *N* donkeys. Initially, the initial position of the donkeys is randomly created in the farm (problem definition space) by (1).

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad (1)$$

Here, X_i is *i-th* donkey, x_i^d is the position d of *i-th* donkey and *n* is the number of dimension of positions.

After the formation of the system, its rules are specified. It is assumed that the donkeys are affected by each other. These donkeys move towards food according to the shortest path.

Positions of the best and the worst donkeys are specified by (2) and (3).

$$Donkey_{best} = location\ of\ \min(fit_j) \quad \& \quad j \in \{1:N\} \quad (2)$$

$$Donkey_{worst} = location\ of\ \max(fit_j) \quad \& \quad j \in \{1:N\} \quad (3)$$

where, fit_j is the value of the fitness function of the *j_th* donkey and *N* is the number of donkeys.

In order to simulate the amount of food in the position of each donkey, a fitness function is used. This parameter is calculated in (4).

$$food_i = \frac{fit_i - fit\ (Donkey_{worst})}{\sum_{j=1}^{N} fit_j - fit\ (Donkey_{worst})} \quad (4)$$

Here $food_i$ is the value of food in location of *i-th* donkey.

For each donkey, locations of food are assumed in (5).

$$candide_i^{food} = \{X_{i,k}^c | X_{i,k}^c \in X, food_{i,k}^c > food_i\} \quad (5)$$

Here $candide_i^{food}$ are locations of food for *i-th* donkey, $X_{i,k}^c$ is the position of *k_th* Candidate of food for the *i_th* donkey.

The distance between the donkey and the place of food is calculated in (6).

$$D_{i,k} = \left(\sum_{j=1}^{n} \left(x_i^j - x_{i,k}^{k,j}\right)^2\right)^{\frac{1}{2}} \quad (6)$$

Here $D_{i,k}$ is the Euclidean distance between the *i-th* donkey and the *k-th* place of food for the *i-th* donkey.

It is assumed that each donkey chooses the shortest path to reach the food. Of course, a donkey may move toward the most food. This process is modeled in (7).

$$X^{i,d} = \begin{cases} X_0^{i,d} + r_1\left(food_i^{select,d} - X_0^{i,d}\right) & if\ P > rand \\ X_0^{i,d} + r_1\left(Donkey_{best}^d - X_0^{i,d}\right) & else \end{cases} \quad (7)$$

where $food_i^{select}$ is the closest food to *i-th* donkey, $r_1$is a random number with normal distribution in the interval [0-1] and *P* is the probability of choosing the shortest path.

B. Repeat steps of the algorithm, until the stopping condition is satisfied.

At first, each donkey stays randomly at a point of the farm, which is the answer to the problem. At each moment of time, donkeys' positions are evaluated. Then the position of each donkey is updated after the calculation of equations 2 to 7. Stop condition can be determined after a certain period of time. The various steps of Donkey Tearoom optimization are as follows:

Start

a) Set up the system.

b) The initial position of the donkeys.

c) Evaluation of players.

d) $Donkey_{best}$ and $Donkey_{worst}$ update.

e) Calcaluation of foods in varios position.

f) Specifition locations of food for each donkey.

g) Calcluation of distance between the donkeys and the place of foods.

h) Update of the position of donkeys.

i) As long as the stop condition is not satisfied, repeat steps c to h.

end

IV. SIMULATION AND RESULT

Performance of the proposed algorithm is evaluated by applying the standard benchmark test functions shown in Table I-III [21]. In order to demonstrate the effectiveness of the proposed algorithm, it is compared with eight well-known optimization algorithms (GA, PSO, GSA, TLBO, WCA, GWO, GOA and EPO) on unimodal, multimodal, fixed-dimension multimodal benchmark test functions.

The experimentation has been done on MATLAB R2014a version in the environment of Microsoft Windows 7 using 64 bit Core i-7 processor with 2.40 GHz and 8 GB main memory. The average and standard deviation of the best optimal solution are mentioned in tables.

TABLE I. UNIMODAL TEST FUNCTIONS

$F_1(x)=\sum_{i=1}^{m} x_i^2$	$[-100,100]^m$
$F_2(x)=\sum_{i=1}^{m} \lvert x_i\rvert + \prod_{i=1}^{m} \lvert x_i\rvert$	$[-10,10]^m$
$F_3(x)=\sum_{i=1}^{m}\left(\sum_{j=1}^{i} x_i\right)^2$	$[-100,100]^m$
$F_4(x)=\max\{\lvert x_i\rvert, \quad 1\le i\le m\}$	$[-100,100]^m$
$F_5(x)=\sum_{i=1}^{m-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2)]$	$[-30,30]^m$
$F_6(x)=\sum_{i=1}^{m}([x_i+0.5])^2$	$[-100,100]^m$
$F_7(x)=\sum_{i=1}^{m} ix_i^4+random(0,1)$	$[-1.28,1.28]^m$

TABLE II. MULTIMODAL TEST FUNCTIONS

$F_8(x)=\sum_{i=1}^{m} -x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-500,500]^m$
$F_9(x)=\sum_{i=1}^{m}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]^m$
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m}x_i^2}\right)-\exp\left(\frac{1}{m}\sum_{i=1}^{m}cos(2\pi x_i)\right)+20+e$	$[-32,32]^m$
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{m}x_i^2-\prod_{i=1}^{m}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^m$
$F_{12}(x)=\frac{\pi}{m}\left\{10\sin(\pi y_1)+\sum_{i=1}^{m}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}+\sum_{i=1}^{m}u(x_i,10,100,4)$ $u(x_i,a,i,n)=\begin{cases}k(x_i-a)^n & x_i>-a\\ 0 & -a< x_i <a\\ k(-x_i-a)^n & x_i<-a\end{cases}$	$[-50,50]^m$
$F_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{m}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2\,[1+\sin^2(2\pi x_m)]\right\}+\sum_{i=1}^{m}u(x_i,5,100,4)$	$[-50,50]^m$

TABLE III. MULTIMODAL TEST FUNCTIONS WITH FIXED DIMENSION

$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	$[-65.53,65.53]^2$
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	$[-5,5]^4$
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$[-5,5]^2$
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	$[-5,10]\times[0,15]$
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$[-5,5]^2$
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-P_{ij})^2)$	$[0,1]^3$
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-P_{ij})^2)$	$[0,1]^6$
$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+6c_i]^{-1}$	$[0,10]^4$
$F_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+6c_i]^{-1}$	$[0,10]^4$
$F_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+6c_i]^{-1}$	$[0,10]^4$

A. Unimodal test functions with high dimension.

Functions F1 to F7 are Unimodal test functions. The average results obtained during 20 times the independent implementation of the algorithms are presented in Table IV. These results indicate that the performance of DTO is better than other algorithms in all F1 to F7 functions.

B. Multimodal test functions with high dimension.

In multimodal test functions, the number of local responses increases exponentially with increasing function dimensions. Therefore, it is hardly possible to achieve the minimum answer in this type of functions. In this type of functions, reaching the nearest answer indicates the high power of the algorithm in passing the wrong local answers. The results of evaluating functions F8 to F13 for 20 independent runtimes are shown in Table V. In all of these functions, DTO has a better performance.

C. Multimodal test functions with low dimension.

Functions F14 to F23 have a low number of dimensions and they also have low local answers. Results of the 20-time implementation of DTO and other algorithms are presented in Table VI. These results show the proper performance of DTO in these types of functions.

TABLE IV. RESULTS FOR DTO AND OTHER ALGORITHMS IN UNIMODAL TEST FUNCTIONS

		GA	PSO	GSA	TLBO	WCA	GWO	GOA	EPO	DTO
F_1	Ave	1.95E-12	4.98E-09	1.16E-16	3.55E-02	2.81E-01	4.61E-23	7.86E-10	5.71E-28	6.74E-35
	std	2.01E-11	1.40E-08	6.10E-17	1.06E-01	1.11E-01	7.37E-23	8.11E-09	8.31E-29	9.17E-36
F_2	Ave	6.53E-18	7.29E-04	1.70E-01	3.23E-05	3.96E-01	1.20E-34	5.99E-20	6.20E-40	7.78E-45
	std	5.10E-17	1.84E-03	9.29E-01	8.57E-05	1.41E-01	1.30E-34	1.11E-17	3.32E-40	3.48E-45
F_3	Ave	7.70E-10	1.40E+01	4.16E+02	4.91E+03	4.31E+01	1.00E-14	9.19E-05	2.05E-19	2.63E-25
	std	7.36E-09	7.13E+00	1.56E+02	3.89E+03	8.97E+00	4.10E-14	6.16E-04	9.17E-20	9.83E-27
F_4	Ave	9.17E+01	6.00E-01	1.12E+00	1.87E+01	8.80E-01	2.02E-14	8.73E-01	4.32E-18	4.65E-26
	std	5.67E+01	1.72E-01	9.89E-01	8.21E+00	2.50E-01	2.43E-14	1.19E-01	3.98E-19	4.68E-29
F_5	Ave	5.57E+02	4.93E+01	3.85E+01	7.37E+02	1.18E+02	2.79E+01	8.91E+02	5.07E+00	5.41E-01
	std	4.16E+01	3.89E+01	3.47E+01	1.98E+03	1.43E+02	1.84E+00	2.97E+02	4.90E-01	5.05E-02
F_6	Ave	3.15E-01	9.23E-09	1.08E-16	4.88E+00	3.15E-01	6.58E-01	8.18E-17	7.01E-19	8.03E-24
	std	9.98E-02	1.78E-08	4.00E-17	9.75E-01	9.98E-02	3.38E-01	1.70E-18	4.39E-20	5.22E-26
F_7	Ave	6.79E-04	6.92E-02	7.68E-01	3.88E-02	2.02E-02	7.80E-04	5.37E-01	2.71E-05	3.33E-08
	std	3.29E-03	2.87E-02	2.77E+00	5.79E-02	7.43E-03	3.85E-04	1.89E-01	9.26E-06	1.18E-06

TABLE V. RESULTS FOR DTO AND OTHER ALGORITHMS IN MULTIMODAL TEST FUNCTIONS

		GA	PSO	GSA	TLBO	WCA	GWO	GOA	EPO	DTO
F_8	Ave	-5.11E+02	-5.01E+02	-2.75E+02	-3.81E+02	-6.92E+02	-6.14E+02	-4.69E+01	-8.76E+02	-1.2E+04
	std	4.37E+01	4.28E+01	5.72E+01	2.83E+01	9.19EE+01	9.32E+01	3.94E+01	5.92E+01	9.14E-12
F_9	Ave	1.23E-01	1.20E-01	3.35E+01	2.23E+01	1.01E+02	4.34E-01	4.85E-02	6.90E-01	8.76E-04
	std	4.11E+01	4.01E+01	1.19E+01	3.25E+01	1.89E+01	1.66E+00	3.91E+01	4.81E-01	4.85E-02
F_{10}	Ave	5.31E-11	5.20E-11	8.25E-09	1.55E+01	1.15E+00	1.63E-14	2.83E-08	8.03E-16	8.04E-20
	std	1.11E-10	1.08E-10	1.90E-09	8.11E+00	7.87E-01	3.14E-15	4.34E-07	2.74E-14	3.34E-18
F_{11}	Ave	3.31E-06	3.24E-06	8.19E+00	3.01E-01	5.74E-01	2.29E-03	2.49E-05	4.20E-05	4.23E-10
	std	4.23E-05	4.11E-05	3.70E+00	2.89E-01	1.12E-01	5.24E-03	1.34E-04	4.73E-04	5.11E-07
F_{12}	Ave	9.16E-08	8.93E-08	2.65E-01	5.21E+01	1.27E+00	3.93E-02	1.34E-05	5.09E-03	6.33E-05
	std	4.88E-07	4.77E-07	3.14E-01	2.47E+02	1.02E+00	2.42E-02	6.23E-04	3.75E-03	4.71E-04
F_{13}	Ave	6.39E-02	6.26E-02	5.73E-32	2.81E+02	6.60E-02	4.75E-01	9.94E-08	1.25E-08	0.00E+00
	std	4.49E-02	4.39E-02	8.95E-32	8.63E+02	4.33E-02	2.38E-01	2.61E-07	2.61E-07	0.00E+00

TABLE VI. RESULTS FOR DTO AND OTHER ALGORITHMS IN MULTIMODAL TEST FUNCTIONS WITH LOW DIMENSION

		GA	PSO	GSA	TLBO	WCA	GWO	GOA	EPO	DTO
F_{14}	Ave	4.39E+00	2.77E+00	3.61E+00	6.79E+00	9.98E+01	3.71E+00	1.26E+00	1.08E+00	9.98E-01
	std	4.41E-02	2.32E+00	2.96E+00	1.12E+00	9.14E-1	3.86E+00	6.86E-01	4.11E-02	7.64E-12
F_{15}	Ave	7.36E-02	9.09E-03	6.84E-02	5.15E-02	7.15E-02	3.66E-02	1.01E-02	8.21E-03	3.3E-04
	std	2.39E-03	2.38E-03	7.37E-02	3.45E-03	1.26E-01	7.60E-02	3.75E-03	4.09E-03	1.25E-05
F_{16}	Ave	-1.02E+00	-1.02E+00	-1.02E+00	-1.01E+00	-1.02E+00	-1.02E+00	-1.02E+00	-1.02E+00	-1.03E+00
	std	4.19E-07	0.00E+00	0.00E+00	3.64E-08	4.74E-08	7.02E-09	3.23E-05	9.80E-07	5.12E-10
F_{17}	Ave	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	std	3.71E-17	9.03E-16	1.13E-16	9.45E-15	1.15E-07	7.00E-07	7.61E-04	5.39E-05	4.56E-21
F_{18}	Ave	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	std	6.33E-07	6.59E-05	3.24E-02	1.94E-10	1.48E+01	7.16E-06	2.25E-05	1.15E-08	1.15E-18
F_{19}	Ave	-3.81E+00	-3.80E+00	-3.86E+00	-3.73E+00	-3.77E+00	-3.84E+00	-3.75E+00	-3.86E+00	-3.86E+00
	std	4.37E-10	3.37E-15	4.15E-01	9.69E-04	3.53E-07	1.57E-03	2.55E-03	6.50E-07	5.61E-10
F_{20}	Ave	-2.39E+00	-3.32E+00	-1.47E+00	-2.17E+00	-3.23E+00	-3.27E+00	-2.84E+00	-2.81E+00	-3.31E+00
	std	4.37E-01	2.66E-01	5.32E-01	1.64E-01	5.37E-02	7.27E-02	3.71E-01	7.11E-01	4.29E-05
F_{21}	Ave	-5.19E+00	-7.54E+00	-4.57E+00	-7.33E+00	-7.38E+00	-9.65E+00	-2.28E+00	-8.07E+00	-10.15E+00
	std	2.34E+00	2.77E+00	1.30E+00	1.29E+00	2.91E+00	1.54E+00	1.80E+00	2.29E+00	1.25E-02
F_{22}	Ave	-2.97E+00	-8.55E+00	-6.58E+00	-1.00E+00	-8.50E+00	-1.04E+00	-3.99E+00	-10.01E+00	-10.40E+00
	std	1.37E-02	3.08E+00	2.64E+00	2.89E-04	3.02E+00	2.73E-04	1.99E+00	3.97E-02	3.65E-07
F_{23}	Ave	-3.10E+00	-9.19E+00	-9.37E+00	-2.46E+00	-8.41E+00	-1.05E+01	-4.49E+00	-3.41E+00	-10.53E+00
	std	2.37E+00	2.52E+00	2.75E+00	1.19E+00	3.13E+00	1.81E-04	1.96E+00	1.11E-02	5.26E-06

V. CONCLOUSION

Meta-heuristic optimization algorithms have been widely used for optimization in recent years. Most of these algorithms are inspired by physical processes and living beings' behavior. In this paper, a new optimization algorithm called Donkey Theorem Optimization based on the Donkey Theorem is introduced. In this Theorem, Donkeys chooses the shortest path to reach the food.

DTO has been tested on 23 benchmark test functions. The results show that DTO provides very competitive results as compared with other well-known metaheuristics such as GA, PSO, GSA, TLBO, WCA, GWO, GOA, and EPO. The results on the unimodal and multimodal test functions show the superior exploitation and exploration capability of DTO.

REFERENCES

[1] R. K. Chandrawat, R. Kumar, B. Garg, G. Dhiman, and S. Kumar, "An Analysis of Modeling and Optimization Production Cost Through Fuzzy Linear Programming Problem with Symmetric and Right Angle Triangular Fuzzy Number," in Proceedings of Sixth International Conference on Soft Computing for Problem Solving, 2017, pp. 197-211.

[2] P. Singh and G. Dhiman, "A fuzzy-lp approach in time series forecasting," in International Conference on Pattern Recognition and Machine Intelligence, 2017, pp. 243-253.

[3] M. Dehghani, Z. Montazeri, A. Ehsanifar, A. Seifi, M. Ebadi, and O. Grechko, "PLANNING OF ENERGY CARRIERS BASED ON FINAL ENERGY CONSUMPTION USING DYNAMIC PROGRAMMING AND PARTICLE SWARM OPTIMIZATION," Електротехніка і Електромеханіка, pp. 62-71, 2018.

[4] Z. Montazeri and T. Niknam, "OPTIMAL UTILIZATION OF ELECTRICAL ENERGY FROM POWER PLANTS BASED ON FINAL ENERGY CONSUMPTION USING GRAVITATIONAL SEARCH ALGORITHM," Электротехника и электромеханика, 2018.

[5] P. Singh and G. Dhiman, "Uncertainty representation using fuzzy-entropy approach: special application in remotely sensed high resolution satellite images (RSHRSIs)," Appl. Soft Comput, 2018.

[6] S. Mirjalili, "Genetic Algorithm," in Evolutionary Algorithms and Neural Networks, ed: Springer, 2019, pp. 43-55.

[7] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, pp. 341-359, 1997.

[8] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–A comprehensive introduction," Natural computing, vol. 1, pp. 3-52, 2002.

[9] D. Simon, "Biogeography-based optimization," IEEE transactions on evolutionary computation, vol. 12, pp. 702-713, 2008.

[10] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," Statistics and computing, vol. 4, pp. 87-112, 1994.

[11] S. Mirjalili, "Particle Swarm Optimisation," in Evolutionary Algorithms and Neural Networks, ed: Springer, 2019, pp. 15-31.

[12] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in International fuzzy systems association world congress, 2007, pp. 789-798.

[13] G. Dhiman and V. Kumar, "Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems," Knowledge-Based Systems, 2018.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, 2014.

[15] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," Advances in Engineering Software, vol. 105, pp. 30-47, 2017.

[16] M. Dehghani, Z. Montazeri, A. Dehghani, and A. Seifi, "Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law," in Knowledge-Based Engineering and Innovation (KBEI), 2017 IEEE 4th International Conference on, 2017, pp. 0210-0214.

[17] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," Information sciences, vol. 179, pp. 2232-2248, 2009.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," science, vol. 220, pp. 671-680, 1983.

[19] R. A. Formato, "Central force optimization: a new metaheuristic with applications in applied electromagnetics," Progress In Electromagnetics Research, vol. 77, pp. 425-491, 2007.

[20] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," Acta Mechanica, vol. 213, pp. 267-289, 2010.

[21] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," Evolutionary Computation, IEEE Transactions on, vol. 3, pp. 82-102, 1999.