



## MLO: Multi Leader Optimizer

**Mohammad Dehghani<sup>1\*</sup>**    **Zeinab Montazeri<sup>1</sup>**    **Ali Dehghani<sup>2</sup>**  
**Ricardo A. Ramirez-Mendoza<sup>3</sup>**    **Haidar Samet<sup>4</sup>**    **Josep M. Guerrero<sup>5</sup>**    **Gaurav Dhiman<sup>6</sup>**

<sup>1</sup>Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, I.R. Iran

<sup>2</sup>Department of Civil Engineering Islamic Azad Universities of Estahban Estahban, I.R. Iran

<sup>3</sup>Tecnológico de Monterrey, Monterrey NL, 64,489, Mexico

<sup>4</sup>Department of Power and Control Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

<sup>5</sup>Center for Research on Microgrids (CROM), Department of Energy Technology, Aalborg University, Aalborg, Denmark

<sup>6</sup>Department of Computer Science, Government Bikram College of Commerce, Patiala, Punjab 147004, India

\* Corresponding author's Email: m.dehghani@sutech.ac.ir

---

**Abstract:** Optimization is a topic that has always been discussed in all different fields of science. One of the most effective techniques for solving such problems is optimization algorithms. In this paper, a new optimizer called Multi-Leader optimizer (MLO) is developed in which multiple leaders guide members of the population towards the optimal answer. MLO is mathematically modelled based on the process of advancing members of the population and following the leaders. MLO performance in optimization is examined on twenty-three standard objective functions. The results of this optimization are compared with the results of the other eight existing optimization algorithms including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Teaching-Learning-Based Optimization (TLBO), Gray Wolf Optimizer (GWO), Grasshopper Optimization Algorithm (GOA), Emperor Penguin Optimizer (EPO), Shell Game Optimization (SGO), and Hide Objects Game Optimization (HOGO). Based on the analysis of the simulation results on unimodal test functions to evaluate exploitation ability and multimodal test functions in order to evaluate exploration ability, it has been determined that MLO has a higher ability to solve optimization problems than existing optimization algorithms.

**Keywords:** Optimization, Optimizer, Multi leader optimizer, Swarm-based optimization, Optimization algorithms.

---

### 1. Introduction

Each optimization problem is modeled in three main parts: primary objectives (constraints), secondary objectives (objective functions), and decision variables (problem variables). Primary objectives, which are the most important part of an optimization problem, include limitations and constraints that should be the first priority in solving the problem. Secondary objectives are the objective functions of the problem that should actually be the minimum or maximum objective based on the conditions required in the problem. Achieving secondary goals must be done by considering the

primary objectives and observing all the constraints and limitations.

After mathematical modeling of an optimization problem, it must be solved using the appropriate method. There are different methods for solving optimization problems. Optimization algorithms with high power in solving optimization problems are always considered as one of the effective methods of solving optimization problems [1]. In this regard, optimization algorithms have been applied by scientists in various fields such as energy [2], protection [3], distribution systems [4, 5], storage designing [6], electrical engineering [7, 8], energy commitment [9], and energy carriers [10, 11] to achieve the optimal solution. Optimization

algorithms with random search in problem solving space can provide a suitable solution for a problem [12].

Over the years, scientists have developed numerous optimization algorithms to solve optimization problems. The basis of the design of optimization algorithms is the attitude to various phenomena in the world, the movements and collective lives of creatures, physical laws, and the rules of individual and group games. In all of these phenomena there is a process of progress and a desire to achieve a goal that is mathematically simulated [13]. Therefore, optimization algorithms can be classified into four general groups of game-based algorithms, physics-based algorithms, evolutionary-based algorithms, and swarm-based algorithms based on their design ideas.

Game-based optimization algorithms are designed based on simulating player behavior and different game rules of both group and individual types. The darts game optimizer (DGO) [14] as one of this types algorithms is designed based on simulation of the darts throws and earning points by players in darts game. Simulation of the performance of football clubs in the football game to achieve the championship in the football league is used to design football game based optimization (FGBO) [13]. Other algorithms in this group include: Orientation Search Algorithm (OSA) [15, 16] based on simulation of orientation game, Dice Game Optimizer (DGO) [17] based on simulation of dice game, Shell Game Optimization (SGO) [18], and Hide Objects Game Optimization (HOGO) [19].

Physics-based optimization algorithms based on mathematical simulation of physical laws and processes are introduced in optimization science. The spring search algorithm (SSA) [12, 20], which is one of the physics-based algorithms, has been developed based on Hooke's law in a system called weight and spring. The laws of physics have provided great capacity for the introduction of new algorithms. Black hole phenomenon is applied to design Black Hole (BH) [21], chemical reactions processes is applied to design Artificial Chemical Reaction Optimization Algorithm (ACROA) [22]. Other algorithms in this group include: Curved Space Optimization (CSO) [23], Galaxy-based Search Algorithm (GbSA) [24], Ray Optimization (RO) [25] algorithm, Small World Optimization Algorithm (SWOA) [26].

Evolutionary-based optimization algorithms have involved evolution of a population in order to create new generations of genetically superior individuals [27]. Differential Evolution (DE) [28], Genetic Algorithm (GA) [29], Genetic Programming (GP)

[30], Evolution Strategy (ES) [31], and Biogeography-based Optimizer (BBO) [32] are part of this group of algorithms.

Swarm-based optimization algorithms have been developed based on simulations of behaviors, lifestyles, and processes in the life of living organisms to solve optimization problems. The behavior of donkeys to achieve feed and donkey theorem has been used in the design of the Donkey Theorem Optimization (DTO) [33]. Simulation of the behavior and movement of ants to achieve the shortest path has been used in the design of the Ant Colony Optimization (ACO) [34]. Other algorithms in this group include: Particle Swarm Optimization (PSO) [35], Group Optimization (GO) [36], Artificial Bee Colony (ABC) [37], Rat Swarm Optimizer (RSO) [38], Bat-inspired Algorithm (BA) [39], Following Optimization Algorithm (FOA) [40], Emperor Penguin Optimizer (EPO) [41], Cuckoo Search (CS) [42], Spotted Hyena Optimizer (SHO) [43], Teaching-Learning-Based Optimization (TLBO) Algorithm [44], Grasshopper Optimization Algorithm (GOA) [45], Doctor and Patient Optimization (DPO) [46], and Gray Wolf Optimizer (GWO) [47].

As mentioned so far, many optimization algorithms based on different ideas to solve various optimization problems have been developed in different sciences. Although the idea of designing optimization algorithms is different, all of these methods provide a solution to the problem based on a random search in the problem-solving space. Therefore, the solution obtained using optimization algorithms is not necessarily the optimal solution. Given that the solution obtained by the algorithm is close to the optimal solution, it is introduced as a quasi-optimal solution. Accordingly, the main criterion for the superiority of optimization algorithms over each other is to provide the best quasi-optimal answer. This has been the main reason for the design of many optimization algorithms by researchers.

Based on this, the contributions the authors of this article are in designing a new algorithm called Multi-Leader optimizer (MLO) algorithm that can be used to solve various optimization problems. The proposed optimizer is designed to guide the algorithm toward a quasi-optimal solution, using more information from members of the population, and following several leaders. The main advantage and feature of the MLO is the lack of control parameters and thus parameter adjustment. Another feature of the proposed algorithm is scanning the search space under the guidance of several leaders instead of just the best member of the population.

The rest of the present paper is as follows: In Section 2, the problem definition is presented. Then, in section 3, the proposed multi-leader optimization algorithm and its mathematical equations are presented. The simulation of proposed algorithm and the discussion of the results are covered in Section 4. Finally, conclusions, summaries and several suggestions for future work are presented.

## 2. Problem definition

Optimization means finding the best possible solution to a problem by considering its constraints. In different fields of science, after facing an optimization problem, first the variables are determined, its different constraints are considered. Then, based on the intended objectives, the objective function is mathematically modeled.

Thus, the mathematical model for an optimization problem is defined using Eq. (1) to Eq. (4).

$$OF = \min \{f(x_1, \dots, x_i, \dots, x_m)\} \quad (1)$$

Subject to

$$g_j(x_1, \dots, x_i, \dots, x_m) \leq 0, \quad j = 1: n_g \quad (2)$$

$$h_k(x_1, \dots, x_i, \dots, x_m) = 0, \quad k = 1: n_h \quad (3)$$

$$x_i \in D_i, \quad i = 1: m \quad (4)$$

Where,  $OF$  is the objective function,  $x_i$  is the  $i$ 'th variable,  $m$  is the number of variables,  $g_j$  is the  $j$ 'th inequality constraint,  $n_g$  is the number of inequality constraint,  $h_k$  is the  $k$ 'th equality constraint,  $n_h$  is the number of equality constraint of optimization problem, and  $D_i$  is the  $i$ 'th domain contains the finite set of values which can be assigned to variable  $x_i$ .

After mathematical modeling of the optimization problem, the problem can be solved using different optimization methods specially optimization algorithms.

Various methods have been proposed to solve optimization problems.

Graphical methods find the answers to the optimization problem by drawing the feasible area and the objective function. These methods are used for problems with two variables. Because the objective and possible function of problems with more than two decision variables can not be drawn and visualized. Graphical methods are rarely used in practice [48]. Classical methods as one of the old methods are able to achieve the optimal answer. These methods, which use gradient information, are used only for continuous problems and cannot be used for discrete problems. With increasing variables

and constraints, it becomes very difficult to use gradient-based methods. Therefore, these methods do not have the necessary efficiency in solving complex problems [49]. Numerical methods for solving optimization problems require an initial conjecture and then improve this conjecture with consecutive repetitions to get close enough to an answer to the problem. In other words, numerical methods based on initial conjecture create a sequence that converges to a solution to a problem. The difference between numerical methods is in how the sequence is created. This is the main weakness of numerical methods, because it is not easy to find the initial conjecture for which the method converges [50].

Optimization algorithms are one of the most widely used methods in solving optimization problems. Optimization algorithms that are designed based on various phenomena and laws of nature by random search in the problem-solving space provide a suitable solution close to the global solution of the problem [19].

The high ability of optimization algorithms to solve optimization problems has led researchers to design a variety of algorithms.

## 3. Multi-Leader optimizer (MLO)

In this section, multi-leader optimizer (MLO) is mathematically modeled for implementation on various optimization problems. The main idea of the proposed algorithm is to use more information from the members of the algorithm population. In this case, members of the algorithm population use the information of several leaders to search in the problem-solving space.

Each member of the algorithm population is a proposed solution to the optimization problem which is defined by Eq. (5).

$$X_i = [x_i^1, \dots, x_i^d, \dots, x_i^m] \quad (5)$$

Where,  $X_i$  is the  $i$ 'th member of population and  $x_i^d$  is the  $d$ 'th variable of optimization problem.

Members of the population are evaluated by placing them in the fitness function. Then the population matrix is sorted based on the smallest value of the fitness function which is defined using Eq. (6).

$$X^{sort} = \left[ \begin{array}{c|c|c} X_1^{sort} & X_{r_1} & \min(fit) \\ \vdots & \vdots & \vdots \\ X_N^{sort} & X_{r_N} & \max(fit) \end{array} \right] \quad (6)$$

Where,  $X^{sort}$  is the sorted matrix of algorithm population,  $X_1^{sort}$  is the member with best fitness

value,  $X_N^{sort}$  is the member with worst fitness value,  $X_{r_1}$  is the member of population with first rank based on fitness value,  $X_{r_N}$  is the member of population with worst rank based on fitness value,  $fit$  is the fitness value, and  $N$  is the number of member of population.

After sorting the population matrix, a certain number of the highest ranked population is selected as the leader. These leaders, which are updated with each iteration, guide members of the population toward the optimal solution. These leaders are determined using (7).

$$L = \{X_l^{sort}, X_l^{sort} \in X^{sort}, l = 1:n_L\} \quad (7)$$

$$X^{sort} = \begin{bmatrix} X_1^{sort} \\ \vdots \\ X_{n_L}^{sort} \\ \vdots \\ X_N^{sort} \end{bmatrix}_{N \times m} \rightarrow L = \begin{bmatrix} X_1^{sort} \\ \vdots \\ X_l^{sort} \\ \vdots \\ X_{n_L}^{sort} \end{bmatrix}_{n_L \times m}$$

Where,  $L$  is the selected leader members matrix and  $n_L$  is the number of leaders.

The algorithm population in MLO is updated in two phases.

In the first phase, each member of the population is moved in the search space based on the position of the leaders. The update method for each member is such that for each variable (each dimension of location) one of the leaders is determined based on the roulette wheel. Each leader may be selected to update multiple variables of each suggestion solution (each population member). The first phase is simulated using Eq. (8) to Eq. (13).

$$fit_i^n = \frac{fit_i - \max(fit)}{\sum_{j=1}^N (fit_j - \max(fit))} \quad (8)$$

$$P_l = \frac{fit_l^n}{\sum_{j=1}^{n_L} fit_j^n} \quad (9)$$

$$C_l = P_l + C_{l-1}, C_0 = 0 \text{ \& } l = 1:n_L \quad (10)$$

$$L_{i,c}^d = \begin{cases} L_1 = X_1^{sort}, & 0 \leq r \leq C_1 \\ \vdots, & \vdots \\ L_c = X_c^{sort}, & C_{c-1} \leq r \leq C_c \\ \vdots, & \vdots \\ L_{n_L} = X_{n_L}^{sort}, & C_{n_L-1} \leq r \leq C_{n_L} \end{cases} \quad (11)$$

$$x_{i,new}^d = x_i^d + rand(L_{i,c}^d - 2 \times x_i^d) \quad (12)$$

$$X_i = \begin{cases} X_{i,new}, & fit_{i,new} \leq fit_i \\ X_i, & \text{else} \end{cases} \quad (13)$$

Where,  $fit_i^n$  is the normalized fitness function for  $i$ 'th population member,  $P_l$  is the probability of selection of  $l$ 'th leader to guide the variable,  $C_l$  is the cumulative probability of  $l$ 'th leader,  $x_{i,new}^d$  is the new value for  $d$ 'th dimension of  $i$ 'th variable,  $L_{i,c}^d$  is the  $d$ 'th dimension of selected  $c$ 'th leader to guide  $d$ 'th variable of  $i$ 'th population member, and  $r$  is the random number in  $[0,1]$  interval.

In the second phase, after updating according to the first phase, each member of the population makes the small random move in their own neighbourhood. If the new position is more appropriate, the member updates its position to new status, otherwise it will return to its previous position. This phase is mathematically modeled using Eq. (14) and Eq. (15).

$$x_{i,new}^d = x_i^d + 2 \times \left(1 - \frac{t}{T}\right) \times (-0.2 + rand \times 0.4) \times x_i^d \quad (14)$$

$$X_i = \begin{cases} X_{i,new}, & fit_{i,new} \leq fit_i \\ X_i, & \text{else} \end{cases} \quad (15)$$

Where,  $t$  is the  $t$ 'th iteration of algorithm and  $T$  is the maximum number of iteration.

The proposed MLO algorithm is applicable to various optimization problems in all field of science. The flowchart of the implementation of the MLO to optimization problem is shown in Fig. 1.

#### 4. Simulation study and discussion

In this section, in order to evaluate the performance of the MLO in solving optimization problems, the proposed optimizer is implemented on a set of twenty-three different objective functions. These objective functions are categorized into three different groups including unimodal [47], multimodal [38], and fixed-dimension multimodal functions [41]. In order to analyze the ability of the MLO to solve these objective functions, the results obtained from the simulation of the proposed optimizer are compared with the results of the other eight existing optimization algorithms including GA [29], PSO [35], TLBO [44], GWO [47], GOA [45], EPO [41], SGO [18], and HOGO [19].

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 deviations of the best optimal solution are mentioned in Tables 1-3.

#### 4.1 Unimodal objective functions

Objective functions  $F_1$  to  $F_7$  as single-model functions are used to evaluate the power of optimization algorithms in achieving the optimal solution. An algorithm that provides a better solution has higher exploitation power. The results of this simulation, shown in Table 1, indicate the superiority and high ability of the proposed MLO algorithm in this type of objective functions.

#### 4.2 Multimodal objective functions with high dimension

Objective functions  $F_8$  to  $F_{13}$  as multimodal test functions are used to evaluate the power of optimization algorithms in searching the search space. In these objective functions, the number of local solutions increases exponentially with increasing function dimensions. Therefore, the algorithm that provides a better solution has higher exploration power than other algorithms. The results of simulation on  $F_8$  to  $F_{13}$ , which is shown in Table 2, indicate the superiority and high ability of the proposed MLO algorithm to solve multimodal test functions.

#### 4.3 Multimodal objective functions with low dimension

Objective functions  $F_{14}$  to  $F_{23}$  as multimodal test functions with low dimension are used to evaluate the power of optimization algorithms in both of the searching the search space and the providing optimal solution. These functions have a low number of dimensions and have few local minimums. Results of implementation of MLO and other mentioned algorithms are presented in Table 3. These results show the proper performance of MLO to optimize these types of objective functions.

#### 4.4 Discussion and theoretical explanation

Exploration power and exploitation power are two very important indicators in evaluating the quality and ability of optimization algorithms to solve optimization problems. An algorithm must scan the search space accurately in the initial iterations. During the iterations of the algorithm and after a proper search, the algorithm must reach the appropriate optimal solution.

The exploitation power of an algorithm indicates the ability of that algorithm to find the appropriate solution to an optimization problem. This concept is explained by the fact that an algorithm that provides a better quasi-optimal solution has higher

exploitation power. Objective functions  $F_1$  to  $F_7$  are used to analyses and compare the exploitation power index in optimization algorithms. According to Table 1, which shows the optimization results of the proposed MLO algorithm and the other eight compared algorithms for these objective functions, it is clear that the MLO has provided much more appropriate solutions and as a result has higher exploitation power than the other eight algorithms.

Exploration power means the ability to search in the problem-solving space of an optimization problem. This factor is very important in optimization problems with the number of local optimal points. An algorithm that scans the search space more accurately has a higher exploration power and is therefore able to achieve a quasi-optimal solution close to the optimal solution by passing through local optimal solutions. Objective functions  $F_8$  to  $F_{23}$  are used to analyze and compare the exploration power index in optimization algorithms. According to Tables 2 and 3, which show the optimization results of the proposed algorithm and the other eight compared algorithms for these objective functions, it is clear that the proposed MLO algorithm with accurate and powerful search of search space provides much more appropriate solutions by passing local optimal solutions and therefore has higher exploration power than other algorithms.

Therefore, it can be stated that the proposed algorithm has been able to provide more appropriate solutions by maintaining a balance between exploration power and exploitation power and is much more competitive than the other eight optimization algorithms to implement in solving optimization problems.

### 5. Conclusions

There are many optimization problems in different fields of science that must be solved in a proper method. Optimization algorithms have a high capability in solving optimization problems and providing quasi-optimal solutions close to the optimal solution. In this paper, a new optimization algorithm called multi-leader optimizer (MLO) has been presented which the main its idea is to use more information of the top population members as leaders in order to move the population in the search space and achieve to quasi-optimal solution.

The performance of the MLO has been evaluated on a set of different objective functions including unimodal, multimodal, and fixed-dimension multimodal functions. Unimodal objective functions

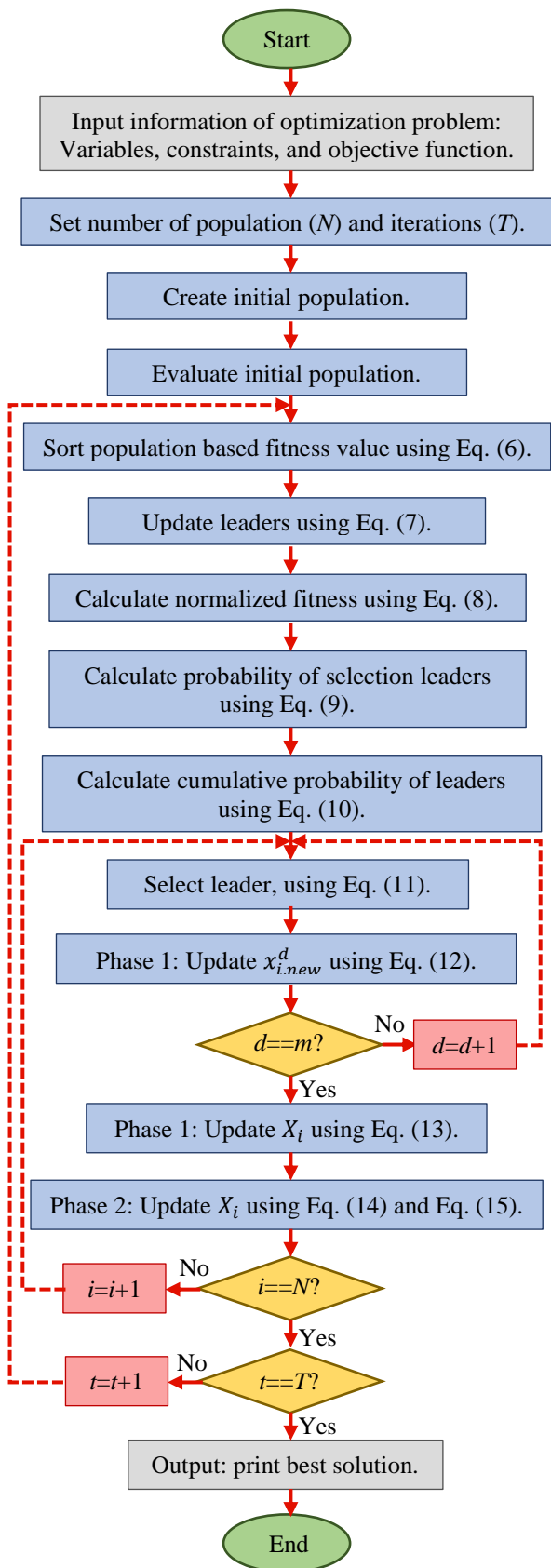


Figure. 1 flowchart of MLO

Table 1. Results for MLO and other algorithms in unimodal test functions

		GA	PSO	TLBO	GWO	GOA	EPO	SGO	HOGO	MLO
F <sub>1</sub>	Ave.	1.95×10 <sup>-12</sup>	4.98×10 <sup>-9</sup>	3.55×10 <sup>-2</sup>	7.86×10 <sup>-10</sup>	2.81×10 <sup>-1</sup>	5.71×10 <sup>-28</sup>	6.74×10 <sup>-35</sup>	4.36×10 <sup>-36</sup>	0
	std	2.01×10 <sup>-11</sup>	1.40×10 <sup>-8</sup>	1.06×10 <sup>-1</sup>	8.11×10 <sup>-9</sup>	1.11×10 <sup>-1</sup>	8.31×10 <sup>-29</sup>	9.17×10 <sup>-36</sup>	6.84×10 <sup>-37</sup>	0
F <sub>2</sub>	Ave	6.53×10 <sup>-18</sup>	7.29×10 <sup>-4</sup>	3.23×10 <sup>-5</sup>	5.99×10 <sup>-20</sup>	3.96×10 <sup>-1</sup>	6.20×10 <sup>-40</sup>	7.78×10 <sup>-45</sup>	5.89×10 <sup>-49</sup>	5.07×10 <sup>-206</sup>
	std	5.10×10 <sup>-17</sup>	1.84×10 <sup>-3</sup>	8.57×10 <sup>-5</sup>	1.11×10 <sup>-17</sup>	1.41×10 <sup>-1</sup>	3.32×10 <sup>-40</sup>	3.48×10 <sup>-45</sup>	3.25×10 <sup>-46</sup>	2.56×10 <sup>-321</sup>
F <sub>3</sub>	Ave	7.70×10 <sup>-10</sup>	1.40×10 <sup>-1</sup>	4.91×10 <sup>+3</sup>	9.19×10 <sup>-5</sup>	4.31×10 <sup>+1</sup>	2.05×10 <sup>-19</sup>	2.63×10 <sup>-25</sup>	6.85×10 <sup>-26</sup>	1.22×10 <sup>-223</sup>
	std	7.36×10 <sup>-9</sup>	7.13	3.89×10 <sup>+3</sup>	6.16×10 <sup>-4</sup>	8.97	9.17×10 <sup>-20</sup>	9.83×10 <sup>-27</sup>	4.52×10 <sup>-29</sup>	5.48×10 <sup>-234</sup>
F <sub>4</sub>	Ave	9.17×10 <sup>+1</sup>	6.00×10 <sup>-1</sup>	1.87×10 <sup>+1</sup>	8.73×10 <sup>-1</sup>	8.80×10 <sup>-1</sup>	4.32×10 <sup>-18</sup>	4.65×10 <sup>-26</sup>	2.15×10 <sup>-27</sup>	1.36×10 <sup>-176</sup>
	std	5.67×10 <sup>+1</sup>	1.72×10 <sup>-1</sup>	8.21	1.19×10 <sup>-1</sup>	2.50×10 <sup>-1</sup>	3.98×10 <sup>-19</sup>	4.68×10 <sup>-29</sup>	5.68×10 <sup>-30</sup>	2.56×10 <sup>-201</sup>
F <sub>5</sub>	Ave	5.57×10 <sup>+2</sup>	4.93×10 <sup>+1</sup>	7.37×10 <sup>+2</sup>	8.91×10 <sup>-2</sup>	1.18×10 <sup>+2</sup>	5.07	5.41×10 <sup>-1</sup>	3.42×10 <sup>-1</sup>	3.45×10 <sup>-1</sup>
	std	4.16×10 <sup>+1</sup>	3.89×10 <sup>+1</sup>	1.98×10 <sup>+3</sup>	2.97×10 <sup>-2</sup>	1.43×10 <sup>+2</sup>	4.90×10 <sup>-1</sup>	5.05×10 <sup>-2</sup>	3.84×10 <sup>-2</sup>	4.95×10 <sup>-4</sup>
F <sub>6</sub>	Ave	3.15×10 <sup>-1</sup>	9.23×10 <sup>-9</sup>	4.88	8.18×10 <sup>-17</sup>	3.15×10 <sup>-1</sup>	7.01×10 <sup>-19</sup>	8.03×10 <sup>-24</sup>	2.69×10 <sup>-26</sup>	0
	std	9.98×10 <sup>-2</sup>	1.78×10 <sup>-8</sup>	9.75×10 <sup>-1</sup>	1.70×10 <sup>-18</sup>	9.98×10 <sup>-2</sup>	4.39×10 <sup>-20</sup>	5.22×10 <sup>-26</sup>	5.62×10 <sup>-28</sup>	0
F <sub>7</sub>	Ave	6.79×10 <sup>-4</sup>	6.92×10 <sup>-2</sup>	3.88×10 <sup>-2</sup>	5.37×10 <sup>-1</sup>	2.02×10 <sup>-2</sup>	2.71×10 <sup>-5</sup>	3.33×10 <sup>-8</sup>	1.85×10 <sup>-9</sup>	1.05×10 <sup>-11</sup>
	std	3.29×10 <sup>-3</sup>	2.87×10 <sup>-2</sup>	5.79×10 <sup>-2</sup>	1.89×10 <sup>-1</sup>	7.43×10 <sup>-3</sup>	9.26×10 <sup>-6</sup>	1.18×10 <sup>-6</sup>	1.05×10 <sup>-7</sup>	5.61×10 <sup>-10</sup>

Table 3. Results for MLOO and other algorithms in multimodal test functions with low dimension

	GA	PSO	TLBO	GWO	GOA	EPO	SGO	HOGO	MLO
F <sub>14</sub>	Ave	4.39	2.77	6.79	1.26	9.98×10 <sup>+1</sup>	1.08	9.98×10 <sup>-1</sup>	6.41×10 <sup>-6</sup>
	std	4.41×10 <sup>-2</sup>	2.32	1.12	6.86×10 <sup>-1</sup>	9.14×10 <sup>-1</sup>	4.11×10 <sup>-2</sup>	7.64×10 <sup>-12</sup>	3.92×10 <sup>-14</sup>
F <sub>15</sub>	Ave	7.36×10 <sup>-2</sup>	9.09×10 <sup>-3</sup>	5.15×10 <sup>-2</sup>	1.01×10 <sup>-2</sup>	7.15×10 <sup>-2</sup>	8.21×10 <sup>-3</sup>	3.3×10 <sup>-4</sup>	1.25×10 <sup>-4</sup>
	std	2.39×10 <sup>-3</sup>	2.38×10 <sup>-3</sup>	3.45×10 <sup>-3</sup>	3.75×10 <sup>-3</sup>	1.26×10 <sup>-1</sup>	4.09×10 <sup>-3</sup>	1.25×10 <sup>-5</sup>	2.15×10 <sup>-5</sup>
F <sub>16</sub>	Ave	-1.02	-1.02	-1.01	-1.02	-1.02	-1.02	-1.03	-1.03
	std	4.19×10 <sup>-7</sup>	0.00	3.64×10 <sup>-8</sup>	3.23×10 <sup>-5</sup>	4.74×10 <sup>-8</sup>	9.80×10 <sup>-7</sup>	5.12×10 <sup>-10</sup>	3.26×10 <sup>-10</sup>
F <sub>17</sub>	Ave	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>	3.98×10 <sup>-1</sup>
	std	3.71×10 <sup>-17</sup>	9.03×10 <sup>-16</sup>	9.45×10 <sup>-15</sup>	7.61×10 <sup>-4</sup>	1.15×10 <sup>-7</sup>	5.39×10 <sup>-5</sup>	4.56×10 <sup>-21</sup>	2.45×10 <sup>-21</sup>
F <sub>18</sub>	Ave	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
	std	6.33×10 <sup>-7</sup>	6.59×10 <sup>-5</sup>	1.94×10 <sup>-10</sup>	2.25×10 <sup>-5</sup>	1.48×10 <sup>+1</sup>	1.15×10 <sup>-8</sup>	1.15×10 <sup>-18</sup>	4.61×10 <sup>-19</sup>
F <sub>19</sub>	Ave	-3.81	-3.80	-3.73	-3.75	-3.77	-3.86	-3.86	-3.86
	std	4.37×10 <sup>-10</sup>	3.37×10 <sup>-15</sup>	9.69×10 <sup>-4</sup>	2.55×10 <sup>-3</sup>	3.53×10 <sup>-7</sup>	6.50×10 <sup>-7</sup>	5.61×10 <sup>-10</sup>	6.23×10 <sup>-11</sup>
F <sub>20</sub>	Ave	-2.39	-3.32	-2.17	-2.84	-3.23	-2.81	-3.31	-3.36
	std	4.37×10 <sup>-1</sup>	2.66×10 <sup>-1</sup>	1.64×10 <sup>-1</sup>	3.71×10 <sup>-1</sup>	5.37×10 <sup>-2</sup>	7.11×10 <sup>-1</sup>	4.29×10 <sup>-5</sup>	2.95×10 <sup>-5</sup>
F <sub>21</sub>	Ave	-5.19	-7.54	-7.33	-2.28	-7.38	-8.07	-10.15	-10.15
	std	2.34	2.77	1.29	1.80	2.91	2.29	1.25×10 <sup>-2</sup>	1.05×10 <sup>-3</sup>
F <sub>22</sub>	Ave	-2.97	-8.55	-1.00	-3.99	-8.50	-10.01	-10.40	-10.40
	std	1.37×10 <sup>-2</sup>	3.08	2.89×10 <sup>-4</sup>	1.99	3.02	3.97×10 <sup>-2</sup>	3.65×10 <sup>-7</sup>	3.25×10 <sup>-8</sup>
F <sub>23</sub>	Ave	-3.10	-9.19	-2.46	-4.49	-8.41	-3.41	-10.53	-10.55
	std	2.37	2.52	1.19	1.96	3.13	1.11×10 <sup>-2</sup>	5.26×10 <sup>-6</sup>	2.91×10 <sup>-6</sup>

Table 2. Results for MLO and other algorithms in multimodal test functions

	GA	PSO	TLBO	GWO	GOA	EPO	SGO	HOGO	MLO
F <sub>8</sub>	Ave	-5.11×10 <sup>+2</sup>	-5.01×10 <sup>+2</sup>	-3.81×10 <sup>+2</sup>	-4.69×10 <sup>+1</sup>	-6.92×10 <sup>+2</sup>	-8.76×10 <sup>+2</sup>	-1.2×10 <sup>+4</sup>	-1.2×10 <sup>+4</sup>
	std	4.37×10 <sup>+1</sup>	4.28×10 <sup>+1</sup>	2.83×10 <sup>+1</sup>	3.94×10 <sup>+1</sup>	9.19×10 <sup>+1</sup>	5.92×10 <sup>+1</sup>	9.14×10 <sup>-12</sup>	8.72×10 <sup>-12</sup>
F <sub>9</sub>	Ave	1.23×10 <sup>-1</sup>	1.20×10 <sup>-1</sup>	2.23×10 <sup>-1</sup>	4.85×10 <sup>-2</sup>	1.01×10 <sup>-2</sup>	6.90×10 <sup>-1</sup>	8.76×10 <sup>-4</sup>	5.62×10 <sup>-4</sup>
	std	4.11×10 <sup>+1</sup>	4.01×10 <sup>+1</sup>	3.25×10 <sup>+1</sup>	3.91×10 <sup>+1</sup>	1.89×10 <sup>+1</sup>	4.81×10 <sup>-1</sup>	4.85×10 <sup>-2</sup>	3.21×10 <sup>-2</sup>
F <sub>10</sub>	Ave	5.31×10 <sup>-11</sup>	5.20×10 <sup>-11</sup>	1.55×10 <sup>-1</sup>	2.83×10 <sup>-8</sup>	1.15	8.03×10 <sup>-16</sup>	8.04×10 <sup>-20</sup>	2.61×10 <sup>-20</sup>
	std	1.11×10 <sup>-10</sup>	1.08×10 <sup>-10</sup>	8.11	4.34×10 <sup>-7</sup>	7.87×10 <sup>-1</sup>	2.74×10 <sup>-14</sup>	3.34×10 <sup>-18</sup>	2.14×10 <sup>-18</sup>
F <sub>11</sub>	Ave	3.31×10 <sup>-6</sup>	3.24×10 <sup>-6</sup>	3.01×10 <sup>-1</sup>	2.49×10 <sup>-5</sup>	5.74×10 <sup>-1</sup>	4.20×10 <sup>-5</sup>	4.23×10 <sup>-10</sup>	1.56×10 <sup>-10</sup>
	std	4.23×10 <sup>-5</sup>	4.11×10 <sup>-5</sup>	2.89×10 <sup>-1</sup>	1.34×10 <sup>-4</sup>	1.12×10 <sup>-1</sup>	4.73×10 <sup>-4</sup>	5.11×10 <sup>-7</sup>	4.15×10 <sup>-7</sup>
F <sub>12</sub>	Ave	9.16×10 <sup>-8</sup>	8.93×10 <sup>-8</sup>	5.21×10 <sup>-1</sup>	1.34×10 <sup>-5</sup>	1.27	5.09×10 <sup>-3</sup>	6.33×10 <sup>-5</sup>	4.87×10 <sup>-5</sup>
	std	4.88×10 <sup>-7</sup>	4.77×10 <sup>-7</sup>	2.47×10 <sup>-2</sup>	6.23×10 <sup>-4</sup>	1.02	3.75×10 <sup>-3</sup>	4.71×10 <sup>-4</sup>	3.96×10 <sup>-4</sup>
F <sub>13</sub>	Ave	6.39×10 <sup>-2</sup>	6.26×10 <sup>-2</sup>	2.81×10 <sup>-2</sup>	9.94×10 <sup>-8</sup>	6.60×10 <sup>-2</sup>	1.25×10 <sup>-8</sup>	0.00	0.00
	std	4.49×10 <sup>-2</sup>	4.39×10 <sup>-2</sup>	8.63×10 <sup>-2</sup>	2.61×10 <sup>-7</sup>	4.33×10 <sup>-2</sup>	2.61×10 <sup>-7</sup>	0.00	0.00

are applied to evaluate the exploitation ability of optimization algorithms. multimodal, and fixed-dimension multimodal objective functions are used in

order to evaluate exploration ability of optimization algorithms.

Based on the results of implementing the proposed algorithm on standard objective functions,

it is determined that the MLO has a high ability to search for problem-solving space as an exploration index and to achieve a quasi-optimal solution very close to the global optimal as an exploitation index. In order to analyse the ability of the MLO to solve these objective functions, the results obtained from the simulation of the proposed optimizer are compared with the results of the other eight existing optimization algorithms including GA, PSO, TLBO, GWO, GOA, EPO, SGO, and HOGO. The results obtained from the simulation and comparison of the results obtained from the MLO with eight other algorithms show the superiority of the proposed algorithm in solving various optimization problems.

### Conflicts of Interest

The authors declare no conflict of interest.

### Author Contributions

Conceptualization, M.D, Z.M, and A.D; methodology, M.D, R.R.M, J.M.G, H.S, and G.D; software, M.D, H.S, G.D, R.R.M; validation, R.R.M, J.M.G, and G.D; formal analysis, A.D, M.D, H.S, and Z.M; investigation, M.D, Z.M, and R.R.M; resources, R.R.M; data curation, A.D, H.S, and G.D; writing—original draft preparation, M.D and Z.M; writing—review and editing, H.S, R.R.M, G.D, and J.M.G; visualization, J.M.G, G.D, H.S, and R.R.M; supervision, M.D; project administration, Z.M; funding acquisition, R.R.M.

### Acknowledgement

The current project was funded by Tecnologicode Monterrey and FEMSA Foundation (grant CAMPUSCITY project).

### References

- [1] S. Kaur, L. K. Awasthi, A. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization", *Engineering Applications of Artificial Intelligence*, Vol. 90, p. 103541, 2020.
- [2] M. Dehghani, Z. Montazeri, and O. P. Malik, "Energy Commitment: A Planning of Energy Carrier Based on Energy Consumption", *Электротехника и электромеханика*, No. 4, pp. 69-72, 2019.
- [3] A. Ehsanifar, M. Dehghani, and M. Allahbakhshi, "Calculating The Leakage Inductance for Transformer Inter-Turn Fault Detection Using Finite Element Method", In: *Proc. of Iranian Conf. on Electrical Engineering (ICEE)*, Tehran, Iran, pp. 1372-1377, 2017.
- [4] M. Dehghani, Z. Montazeri, and O. P. Malik, "Optimal Sizing and Placement of Capacitor Banks and Distributed Generation in Distribution Systems Using Spring Search Algorithm", *International Journal of Emerging Electric Power Systems*, Vol. 21, No. 1, 20190217, 2020.
- [5] M. Dehghani, Z. Montazeri, O. P. Malik, K. Al-Haddad, J. M. Guerrero, and G. Dhiman, "A New Methodology Called Dice Game Optimizer for Capacitor Placement in Distribution Systems", *Электротехника и электромеханика*, No. 1, pp. 61-64, 2020.
- [6] S. Dehbozorgi, A. Ehsanifar, Z. Montazeri, M. Dehghani, and A. Seifi, "Line Loss Reduction and Voltage Profile Improvement in Radial Distribution Networks Using Battery Energy Storage System", In: *Proc. of IEEE 4th International Conf. on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 0215-0219, 2017.
- [7] M. Dehghani, M. Mardaneh, Z. Montazeri, A. Ehsanifar, M. Ebadi, and O. Grechko, "Spring Search Algorithm for Simultaneous Placement of Distributed Generation and Capacitors", *Электротехніка і Електромеханіка*, No. 6, pp. 68-73, 2018.
- [8] 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*, No. 4, pp. 70-73, 2018.
- [9] M. Dehghani, M. Mardaneh, O. P. Malik, J.M. Guerrero, R. Morales-Menendez, R. A. Ramirez-Mendoza., J. Matas, and A. Abusorrah, "Energy Commitment for a Power System Supplied by a Multiple Energy Carriers System using Following Optimization Algorithm", *Applied Sciences*, 2020.
- [10] 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", *Электротехника и электромеханика*, No. 5, pp. 62-71, 2018.
- [11] Z. Montazeri and T. Niknam, "Energy Carriers Management Based on Energy Consumption", In: *Proc. of IEEE 4th International Conf. on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 0539-0543, 2017.



- [12] M. Dehghani, Z. Montazeri, A. Dehghani, N. Nouri, and A. Seifi, "BSSA: Binary Spring Search Algorithm", In: *Proc. of IEEE 4th International Conf. on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp.0220-0224, 2017.
- [13] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, and V. Kumar, "Football Game Based Optimization: an Application to Solve Energy Commitment Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 514-523, 2020.
- [14] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman "Darts Game Optimizer: A New Optimization Technique Based on Darts Game", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 286-294, 2020.
- [15] M. Dehghani, Z. Montazeri, O. P. Malik, A. Ehsanifar, and A. Dehghani, "OSA: Orientation Search Algorithm", *International Journal of Industrial Electronics, Control and Optimization*, Vol. 2, No. 2, pp. 99-112, 2019.
- [16] M. Dehghani, Z. Montazeri, O. P. Malik, G. Dhiman, and V. Kumar, "BOSA: Binary Orientation Search Algorithm", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Vol. 9, pp. 5306-5310, 2019.
- [17] M. Dehghani, Z. Montazeri, and O. P. MALIK, "DGO: Dice Game Optimizer", *Gazi University Journal of Science*, Vol. 32, pp. 871-882, 2019.
- [18] M. Dehghani, Z. Montazeri, O. P. Malik, H. Givi, and J. M. Guerrero, "Shell Game Optimization: A Novel Game-Based Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 246-255, 2020.
- [19] M. Dehghani, Z. Montazeri, S. Saremi, A. Dehghani, O. P. Malik, K. Al-Haddad, J. M. Guerrero, "HOGO: Hide Objects Game Optimization", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 216-225, 2020.
- [20] M. Dehghani, Z. Montazeri, A. Dehghani, and A. Seifi, "Spring Search Algorithm: A New Meta-Heuristic Optimization Algorithm Inspired by Hooke's Law", In: *Proc. of IEEE 4th International Conf. on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 0210-0214, 2017.
- [21] A. Hatamlou, "Black Hole: A New Heuristic Optimization Approach for Data Clustering", *Information Sciences*, Vol. 222, pp.175-184, 2013.
- [22] B. Alatas, "ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization", *Expert Systems with Applications*, Vol. 38, pp. 13170-13180, 2011.
- [23] F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, "Curved Space Optimization: A Random Search Based on General Relativity Theory", *arXiv preprint arXiv:1208.2214*, 2012.
- [24] H. Shah-Hosseini, "Principal Components Analysis by the Galaxy-Based Search Algorithm: A Novel Metaheuristic for Continuous Optimisation", *International Journal of Computational Science and Engineering*, Vol. 6, pp. 132-140, 2011.
- [25] A. Kaveh and M. Khayatazad, "A New Meta-Heuristic Method: Ray Optimization", *Computers & Structures*, Vol. 112, pp. 283-294, 2012.
- [26] H. Du, X. Wu, and J. Zhuang, "Small-World Optimization Algorithm for Function Optimization", In: *Proc. of International Conf. on Natural Computation*, pp. 264-273, 2006.
- [27] N. E. Karkalos, A. P. Markopoulos, and J. P. Davim, "Evolutionary-Based Methods", In: *Computational Methods for Application in Industry 4.0*, ed: Springer, pp. 11-31, 2019.
- [28] R. Storn and K. Price, "Differential Evolution-A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces [r]", *Berkeley: ICSI*, 1995.
- [29] K.-S. Tang, K.-F. Man, S. Kwong, and Q. He, "Genetic Algorithms and Their Applications", *IEEE Signal Processing Magazine*, Vol. 13, pp. 22-37, 1996.
- [30] J. R. Koza, "Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems", Stanford University, Department of Computer Science, 1990.
- [31] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies—A Comprehensive Introduction", *Natural Computing*, Vol. 1, pp. 3-52, 2002.
- [32] S. Mirjalili, "Biogeography-Based Optimisation", In: *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 57-72, 2019.
- [33] M. Dehghani, M. Mardaneh, O. P. Malik, and S. M. NouraeiPour, "DTO: Donkey Theorem Optimization", In: *Proc. of 27th Iranian Conf. on Electrical Engineering (ICEE)*, Yazd, Iran, pp. 1855-1859, 2019.
- [34] M. Dorigo and T. Stützle, "Ant Colony Optimization: Overview and Recent Advances", In: *Handbook of Metaheuristics*, ed: Springer, pp. 311-351, 2019.

- [35] J. C. Bansal, "Particle Swarm Optimization", In: *Evolutionary and Swarm Intelligence Algorithms*, ed: Springer, pp. 11-23, 2019.
- [36] M. Dehghani, Z. Montazeri, A. Dehghani, and O. P. Malik, "GO: Group Optimization", *Gazi University Journal of Science*, Vol. 33, 2020.
- [37] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems", In: *Proc. of International Fuzzy Systems Association World Congress*, pp. 789-798, 2007.
- [38] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A Novel Algorithm for Global Optimization: Rat Swarm Optimizer", *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [39] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, ed: Springer, pp. 65-74, 2010.
- [40] M. Dehghani, M. Mardaneh, and O. P. Malik, "FOA: 'Following' Optimization Algorithm for Solving Power Engineering Optimization Problems", *Journal of Operation and Automation in Power Engineering*, Vol. 8, No. 1, pp. 57-64, 2020.
- [41] G. Dhiman and V. Kumar, "Emperor Penguin Optimizer: A Bio-Inspired Algorithm for Engineering Problems", *Knowledge-Based Systems*, Vol. 159, pp. 20-50, 2018.
- [42] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems", *Engineering with Computers*, Vol. 29, pp. 17-35, 2013.
- [43] G. Dhiman and V. Kumar, "Spotted Hyena Optimizer: A Novel Bio-Inspired Based Metaheuristic Technique for Engineering Applications", *Advances in Engineering Software*, Vol. 114, pp. 48-70, 2017.
- [44] P. Sarzaeim, O. Bozorg-Haddad, and X. Chu, "Teaching-Learning-Based Optimization (TLBO) Algorithm", in *Advanced Optimization by Nature-Inspired Algorithms*, ed: Springer, pp. 51-58, 2018.
- [45] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and Application", *Advances in Engineering Software*, Vol. 105, pp. 30-47, 2017.
- [46] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, R. A. Ramirez-Mendoza, J. Matas, J. C. Vasquez, and L. Parra-Arroyo, "A New 'Doctor and Patient' Optimization Algorithm: An Application to Energy Commitment Problem", *Applied Sciences*, 2020.
- [47] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.
- [48] O. Banerjee, L. E. Ghaoui, A. d'Aspremont, and G. Natsoulis, "Convex optimization techniques for fitting sparse Gaussian graphical models", In: *Proc. of the 23rd International Conf. on Machine Learning*, pp. 89-96, 2006.
- [49] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization", *SIAM Journal on Optimization*, Vol. 2, pp. 21-42, 1992.
- [50] M. Gilli, D. Maringer, and E. Schumann, "Numerical Methods and Optimization in Finance", Academic Press, 2019.