International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

HOGO: Hide Objects Game Optimization

Mohammad Dehghani[1]* Zeinab Montazeri[1] Shahrzad Saremi[2] Ali Dehghani[3]
Om Parkash Malik[4] Kamal Al-Haddad[5] Josep M. Guerrero[6]

[1]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran*
[2]*Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Fortitude Valley, Brisbane, 4006 QLD, Australia*
[3]*Department of Civil Engineering, Islamic Azad Universities of Estahban, Estahban, I.R. Iran*
[4]*Department of Electrical Engineering, University of Calgary, Calgary Alberta, Canada*
[5]*École de Technologie Supérieur University of Quebe Montreal, Canada*
[6]*Center for Research on Microgrids (CROM), Department of Energy Technology, Aalborg University, Aalborg, Denmark*
* Corresponding author's Email: adanbax@gmail.com

Abstract: A number of metaphysical algorithms have been developed in recent years. Most of these algorithms are inspired by physical processes or living beings' behaviour. In this paper, a new algorithm namely "Hide Objects Game Optimization (HOGO)" is presented to obtain quasi-optimal solution. It is inspired by an old game and the searcher agents who try to find a hidden object in a given space. In this game, any player must notice the following points: (a) pay attention to the voices made by the coach for players, (b) get closer to the best player for whom the coach made the loudest voice, (c) take influence from the voices made by the coach for other players, (d) compare the new voice after a move with the old voice before the move and return back in case the voice gets lower. HOGO is tested on 23 well-known benchmark test functions and is compared with eight optimization algorithms: Genetic Algorithm, Particle Swarm Optimization, Gravitational Search Algorithm, Teaching Learning Based Optimization, Grey Wolf Optimizer, Grasshopper Optimization Algorithm, Spotted Hyena Optimizer, and Emperor Penguin Optimizer. The results and data obtained from applying HOGO and other said algorithms show that HOGO is able to provide better results in comparison with other well-known optimization algorithms.

Keywords: Hide objects game optimization, Metaheuristic, Optimization, Game, Power engineering.

1. Introduction

1.1 Motivation

In recent years, meta-heuristic algorithms provide optimal solution for real-life problems than the classical techniques [1-3]. Meta-heuristic algorithms have demonstrated their high ability in many fields like Logistics [4], Bioinformatics [5], Data-mining [6], Chemical physics [7], Power engineering [8, 9], Energy [10, 11], energy carrier [12], protection [13], etc.

The mathematical modelling for meta-heuristic search process is difficult [14]. Population-based methods are derived from social interactions among a typical set of society members [15]. Besides, heuristic search algorithms are formed by nature as well as physical and biological inspirations. Unlike classical methods, heuristic search methods use search space in a random though parallel mode. Another noteworthy difference between classical and heuristic methods is that the latter did not use the space gradient information. Most heuristic algorithms merely use objective function to navigate the search. However, they are able to achieve the optimal results. Indeed, the members' behaviour can internally organize the system by forming some properties/features like the positive feedback, negative feedback, inter-search balance, efficiency, and multiple interactions. The mentioned process is called self-organization [16, 17].

1.2 Contribution

Hitherto, many algorithms have been designed by researchers in physics-based, swarm-based, and evolutionary-based algorithms that are used in various scientific fields. Since players try to attain a target called victory in various group and individual games, the rules of these games are also very useful to design new optimization algorithms.

In this paper, a new game-based algorithm called Hide Objects Game Optimization is proposed that used in designing quasi-optimization algorithms. In the proposed algorithm, rules governing the game, the members' influence on each other, and the coach's influence on members are used to design a quasi-optimization algorithm known as Hide Objects Game Optimization (HOGO) algorithm. Although many optimization algorithms face with setting of multiple control parameters, the lack of control parameters is the important strong point of HOGO.

1.3 Paper structure

A brief history of the heuristics-based quasi-optimization methods is given in section 2. The proposed Hide Objects Game Optimization (HOGO) algorithm is presented in Section 3. Properties of the proposed algorithm are explained in section 4. Experimental results are mentioned in section 5. Conclusions are drawn in section 6.

2. Related works

The most commonly algorithms are Genetic algorithm (GA) [18], Simulated Annealing (SA) [19], Harmony Search (HS) [20], Artificial Immune System (AIS) [21], Ant Colony Optimization (ACO) [22] and Particle Swarm Optimization (PSO) [23]. GA which is derived from the genetic law and reproduction is indeed based on the Darwin's theory [24]. SA is based on the process of cooling metals during metallurgy [19]. HS is an algorithm that imitates the melody improving process by the composer while composing [25]. AIS is inspired by the human body's biological system [21]. ACO simulates behaviour of ants while searching food [22]. PSO is derived from the birds' social behaviour while immigrating [26].

Physics-based algorithm have been developed using the rules of physics. Some of this kind of algorithms are: Spring Search Algorithm (SSA) [27] Inspired by Hooke's law, Gravitation Search Algorithm (GSA) [28] based on gravitational gravity force, Charged System Search (CSS) [29] based on some principles from physics and mechanics which each agent is a Charged Particle, Galaxy-based Search Algorithm (GbSA) [30] based on spiral arm of spiral galaxies, Curved Space Optimization (CSO) [31] based on transformation of a random search space into a new search space based on concepts of space-time curvature in general relativity theory, Ray Optimization (RO) [32] algorithm based on the Snell's light refraction law, Artificial Chemical Reaction Optimization Algorithm (ACROA) [33] based on chemical reactions possess, Small World Optimization Algorithm (SWOA) [34] based on mechanism of small-world phenomenon, and Black Hole (BH) [35] based on black hole phenomenon.

Evolutionary-based algorithms have been involved evolutionary of a population in order to create new generations of genetically superior individuals are presented [36]. Genetic Algorithm (GA) [18], Differential Evolution (DE) [37], Evolution Strategy (ES) [38], Genetic Programming (GP) [39], and Biogeography-based Optimizer (BBO) [40] are part of this group of algorithms.

Swarm-based algorithms is inspired from the natural processes of plants, foraging behaviors of insects and social behaviors of animals. Some of these are Particle Swarm Optimization (PSO) [23], Ant Colony Optimization (ACO) [41], Artificial Bee Colony (ABC) [42], Bat-inspired Algorithm (BA) [43], Spotted Hyena Optimizer (SHO) [44], Cuckoo Search (CS) [45], Emperor Penguin Optimizer (EPO) [46], Grey Wolf Optimizer (GWO) [47], 'Following' Optimization Algorithm (FOA) [48], Orientation Search Algorithm (OSA) [49, 50], Group Optimization (GO) [51], Dice Game Optimizer (DGO) [52], Donkey Theorem Optimization (DTO) [53], Shell Game Optimization (SGO) [54], and Grasshopper Optimization Algorithm (GOA) [55].

In their performance structure, all the above-mentioned algorithms have used a kind of statistical property and randomized phenomena, which exist in nature. In some other central force optimizations, which are indeed the universal gravity law metaphors, these random phenomena are not used and it is said that algorithms like these have the certainty property [56].

3. Hide objects game

The hide objects game is an old game. In detail, at first, the coach gives an object to players; then, players known as searchers go out of the game field till the coach hides the object; after that, players return to the field and search while the coach tries to guide them by making sound/voice in that the voice gets louder when any player comes closer to the object and it gets lower when any player goes farther from the object.

In this game, any player must notice the following points:

- pay attention to the voices made by the coach for players
- get closer to the best player for whom the coach made the loudest voice
- take influence from the voices made by the coach for other players
- compare the new voice after a move with the old voice before the move and return back in case the voice gets lower.

4. Hide objects game optimization

HOGO is defined in two general steps: 1- forming a time discrete artificial system in the problem space, the initial positioning of members, determining the governing laws and arranging parameters, 2- passing the time untill arriving at the stop time.

4.1 Making the system, determining the laws and arranging parameters

Imagine the system as a set of ‘m’ players. The location of each player is a point in the space where it is a solution to the optimization problem. In Eq. (1) the *d* dimension location of the *i*-th player is shown as x_i^d.

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n) \tag{1}$$

At first, the initial location of these game players is made randomly on the game field. These players move towards the hidden object based on the laws that govern the game.

In this algorithm, the locations of the best and the worst players are shown as $player_{best}$ and $player_{worst}$, respectively. These locations are shown in Eqs. (2) and (3).

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

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

In these relations, fit_j shows the value of the objective function member *j* and *N* shows the number of players. As mentioned in section 3, while playing HOGO, each player must take 4 points into account:

- *paying attention to the voices made by the coach for players*

To simulate the voices made by the coach, an objective function has been used. This means that any player with the better location is more suitable and accordingly he/she receives a louder voice. the voice made by the coach, is normalized and then it is computed using Eq. (4).

$$Voice_i = \frac{fit_i - fit\ (player_{worst})}{\sum_{j=1}^{N}[fit_j - fit\ (player_{worst})]} \tag{4}$$

Here, $Voice_i$ is a voice made by the coach for member ‘i’. The accumulation possibility of this voice is computed based on Eq. (5).

$$P_i = \frac{Voice_i}{\sum_{j=1}^{N} Voice_j} \tag{5}$$

- *getting closer to the best player for whom the coach made the loudest voice*

In this game concerning the loudness of the voice, any player tries to guide his/her direction to the player for whom the coach made the loudest voice. This strategy is simulated based on Eq. (6).

$$dX_1^{j,d} = player_{best}^d - X_0^{j,d} \tag{6}$$

Here, $dX_1^{j,d}$ shows the movement value of the j member’s d-th dimension towards the best player location, and $X_0^{j,d}$ represents the initial location of the j member’s d-th dimension.

- *Receding from the worst player for whom the coach made the lowest voice*

In this strategy, players try to move away from the player who has the worst location and the lowest voice. This movement is shown in Eq. (7).

$$dX_2^{j,d} = X_0^{j,d} - player_{worst}^d \tag{7}$$

Here, $dX_2^{j,d}$ shows the movement value of the j member’s d-th dimension from the worst player location.

- *Taking influence from the voices made by the coach for other players*

Rather than the best and the worst players, each player of this game tries to take the best advantage of other players’ locations. Here, each player assesses

Received: March 24, 2020. Revised: April 27, 2020.

the loudness of the voices made for other players and moves toward or away from the others by comparing the loudness of voices. To simulate this strategy, the roulette wheel operator is used. Therefore, the possible accumulation function computed in Eq. (5) is used to determine the influential player. Eq. (8) shows the movement value suitable for this strategy.

$$dX_3^{j,d} = \begin{cases} X_0^{j,d} - X_0^{select,d} & P_j > P_{select} \\ X_0^{select,d} - X_0^{j,d} & else \end{cases} \quad (8)$$

Here, $dX_3^{j,d}$ shows the d-th dimension movement of member j that is suitable for the chosen influential player's location, $X_0^{select,d}$ is the d-th dimension location, and P_{select} is the possible accumulation of the chosen influential player and P_j manifests member j's possible accumulation. Now, $X'^{j,d}$ is the new initial location of the j member's d-th dimension that is calculated based on Eq. (9). here, r_1, r_2, and r_3 are random numbers with normal distributions in $[0-1]$ span.

$$X'^{j,d} = X_0^{j,d} + r_1 \times dX^{j,d}{}_1 + r_2 \times dX^{j,d}{}_2 + r_3 \times dX^{j,d}{}_3 \quad (9)$$

- *Comparing the new voice after the move with the voice prior to the move and return back in case the voice gets lower*

In this strategy, the player is temporarily placed on location X^j based on Eq. (9). In this state suitable to the location, the coach makes a new sound as voicej for the player. Here, it is the player who decides where he/she must stand by comparing the new and the old voice of the coach. This means that the player stands in the new location if the new voice is higher than the old one. Otherwise, the player must return back to his/her previous location though this return may not be fulfilled since he/she exit the previous location. Accordingly, based on Eq. (10), he/she may be on a random location around the previous X^j location and *rand* is random numbers with normal distributions in $[0-1]$ span. This strategy is well determined in Eq. (11).

$$X''^{j,d} = (0.9 + 0.2 \times rand) \times X_0^{j,d} \quad (10)$$

$$X^j = \begin{cases} X'^j & Voice'^j > Voice^j \\ X_0^{j,d} & \dfrac{Voice^j}{\max(Voice)} > 0.5 \\ X''^j & else \end{cases} \quad (11)$$

4.2 Passing the time and updating parameters

At the beginning of forming the system, any player is randomly placed in any location of the game. At any moment of time, players' locations are assessed and then their dislocations are measured based on Eqs. (1) to (11). The stop condition can be determined after the passing of a distinct period of time. Different steps of the HOGO algorithm are as follows:

1. Determining the system atmosphere/space and the initial quantifying
2. Initial locating of players
3. Assessing players
4. Determining the voice for each player by the coach
5. Updating the locations of the best and the worst players
6. Updating the players' locations
7. Repeating steps 3 to 6 till the stop condition is satisfied
8. Finishing

The flowchart of HOGO is shown in Fig. 1.

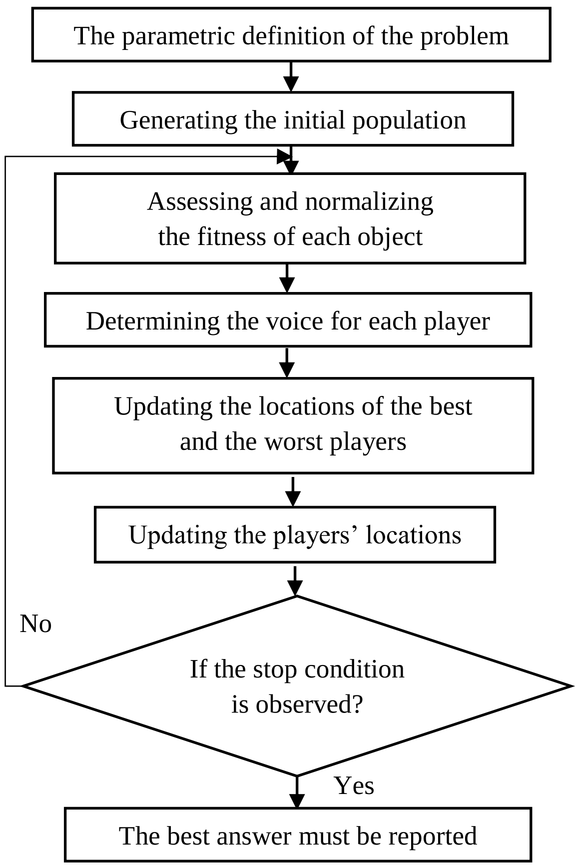

Figure. 1 Hide Object Game Optimization (HOGO) flowchart

5. Experimental results

Performance of the HOGO is assessed by using 23 benchmark test functions [57].

5.1 Algorithms used for comparison

Performance of the HOGO is compared with the following eight optimization algorithms: Genetic Algorithm (GA) [24], Particle Swarm Optimization (PSO) [58], Gravitational Search Algorithm (GSA) [28], Teaching Learning Based Optimization (TLBO) [59], Grey Wolf Optimizer (GWO) [47], Grasshopper Optimization Algorithm (GOA) [55], Spotted Hyena Optimizer (SHO) [44], and Emperor Penguin Optimizer (EPO) [46].

- **Genetic Algorithm (GA) [24]:** GA is based on the survival of the highest and the natural selection of genetic science and Darwin evolutions.
- **Particle Swarm Optimization (PSO) [58]:** PSO is a swarm-based algorithm which simulated the movement of the bird group as part of a sociological study that studies the concept of collective intelligence in the biological community.
- **Gravitational Search Algorithm (GSA) [28]:** GSA is inspired by law of gravity that its search agents are a set of objects that can be thought as planets of a system.
- **Teaching Learning Based Optimization (TLBO) [59]:** TLBO is based on the two phases of teaching and learning. The first stage involves teacher-learning, and the second stage involves learning from one another.
- **Grey Wolf Optimizer (GWO) [47]:** GWO is an algorithm based on nature and social behavior of the wolf during hunting.
- **Grasshopper Optimization Algorithm (GOA) [55]:** GOA is a nature-inspired algorithm that imitates and simulates the behavior of grasshoppers in the nature and the swarm movement of grasshoppers toward food sources.
- **Spotted Hyena Optimizer (SHO) [44]:** SHO is inspired by the behavior of spotted hyenas. The main concept behind this algorithm is the social relationship between spotted hyenas and their collaborative behavior.
- **Emperor Penguin Optimizer (EPO) [46]:** EPO is a swarm-based algorithm which simulates the behavior of the emperor's penguins.

5.2 Evaluation of unimodal test function with high dimensions

Functions F_1 to F_7 are unimodal. The mean results of 20 times of the algorithm’s independent running, are shown in Table 1. These results show that the proposed HOGO has a better performance in all F_1 to F_7 functions than other algorithms.

5.3 Evaluation of multimodal test functions with high dimensions

In multimodal functions of F_8 to F_{13}, by increasing the function dimensions, the number of local responses is increased exponentially. Therefore, arriving at the minimum response of these functions is hardly possible. In these types of functions, arriving at a response close to the ideal response represents the algorithm’s high power in passing through the local wrong responses. Results gained from assessing F8 to F13 after 20 times running of HOGO and other algorithms are shown in Table 2. In all these functions, HOGO shows a better performance.

5.4 Evaluation of multimodal test functions with low dimensions

Functions F_{14} to F_{23} have both low dimensions and low local responses. Results obtained from 20 times running of HOGO and other algorithm, are shown in Table 3. These results represent the suitable performance of HOGO in relation to other algorithms.

5.5 Properties of the proposed algorithm

In optimization, two functions are introduced as exploration and exploitation. In exploration, any optimization algorithm must be well able to search the whole problem space and this search must not be limited to some locations. In exploitation, the algorithm’s ability in exploring optimal locations is the focus. In population-based algorithms, during the initial times of running the algorithm, a comprehensive search of the space is needed and along with the initial repetition, the algorithm must search the space as best as possible. However, as time passes the algorithm’s ability is better revealed and the algorithm must locate optimal points concerning the population’s findings [60].

Concerning the suitable number of members, the above algorithm can well search the problem space. The proposed strategy to improve and hasten the algorithm’s search ability is the influence of players’ dislocations strategy on each other. The mentioned strategy is controlled by Eq. (11). During the initial iterations of this algorithm, the problem still needs the proper search though as time passes, the population arrives at better results.

In any iteration of the algorithm, players are influenced by each other concerning the loud voice receiver, the low voice receiver and other players. During initial iterations, the search space is well analysed not to put the algorithm in the local optimum. Since it is known that, after passing of sometime, players aggregate around better locations and since it is necessary to search the space more precisely with smaller steps, the players' influence on each other is decreased as time passes. Therefore, players are supposed to go to better locations as time passes.

Note that the computational complexity of the propose method is of *O(nmt)* where n is the number of solutions, *m* is the number of variables (dimension), and *t* shows the number of iterations. Therefore, the method can be considered as a computationally cheap method as compared to other meta-heuristics.

6. Conclusion

Heuristics based algorithms have been widely used recently for optimization purposes. Most of these algorithms are formulated by taking inspiration from physical processes or living beings' behaviour. In this article, a new optimization algorithm, known as the hide objects game optimization (HOGO), is proposed. HOGO is introduced based on laws governing a game. In this game, players try to find the hidden object by taking the impression both from the coach and from each other. Two benchmark suites have been used to assess the proposed algorithm. On 23 benchmark test function HOGO and eight additional optimization algorithms were evaluated. HOGO performs well compared to GA, PSO, GSA, TLBO, GWO, GOA, SHO and EPO, according to the results. Based on the results achieved for HOGO and other mentioned optimization algorithms, it has been demonstrated that HOGO can very efficiently manage various kinds of restrictions and provides better solutions. Results obtained from running HOGO show that it performs very satisfactorily for all criterion functions.

In future works, the authors propose several ideas for study. One may create a binary variant of HOGO as an important potential contribution. HOGO may also be used to overcome many-objective real-life optimization as well as multi-objective problems.

Table 1. Results for HOGO and other algorithms considering Unimodal test functions.

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	HOGO
F_1	Ave	1.95×10^{-12}	4.98×10^{-9}	1.16×10^{-16}	3.55×10^{-2}	2.81×10^{-1}	7.86×10^{-10}	4.61×10^{-23}	5.71×10^{-28}	5.32×10^{-36}
	std	2.01×10^{-11}	1.40×10^{-8}	6.10×10^{-17}	1.06×10^{-1}	1.11×10^{-1}	8.11×10^{-9}	7.37×10^{-23}	8.31×10^{-29}	8.24×10^{-37}
F_2	Ave	6.53×10^{-18}	7.29×10^{-4}	1.70×10^{-1}	3.23×10^{-5}	3.96×10^{-1}	5.99×10^{-20}	1.20×10^{-34}	6.20×10^{-40}	6.25×10^{-49}
	std	5.10×10^{-17}	1.84×10^{-3}	9.29×10^{-1}	8.57×10^{-5}	1.41×10^{-1}	1.11×10^{-17}	1.30×10^{-34}	3.32×10^{-40}	2.35×10^{-46}
F_3	Ave	7.70×10^{-10}	$1.40\times10^{+1}$	$4.16\times10^{+2}$	$4.91\times10^{+3}$	$4.31\times10^{+1}$	9.19×10^{-5}	1.00×10^{-14}	2.05×10^{-19}	7.12×10^{-26}
	std	7.36×10^{-9}	7.13	$1.56\times10^{+2}$	$3.89\times10^{+3}$	8.97	6.16×10^{-4}	4.10×10^{-14}	9.17×10^{-20}	5.61×10^{-29}
F_4	Ave	$9.17\times10^{+1}$	6.00×10^{-1}	1.12	$1.87\times10^{+1}$	8.80×10^{-1}	8.73×10^{-1}	2.02×10^{-14}	4.32×10^{-18}	2.14×10^{-27}
	std	$5.67\times10^{+1}$	1.72×10^{-1}	9.89×10^{-1}	8.21	2.50×10^{-1}	1.19×10^{-1}	2.43×10^{-14}	3.98×10^{-19}	6.85×10^{-30}
F_5	Ave	$5.57\times10^{+2}$	$4.93\times10^{+1}$	$3.85\times10^{+1}$	$7.37\times10^{+2}$	$1.18\times10^{+2}$	$8.91\times10^{+2}$	$2.79\times10^{+1}$	5.07	4.32×10^{-1}
	std	$4.16\times10^{+1}$	$3.89\times10^{+1}$	$3.47\times10^{+1}$	$1.98\times10^{+3}$	$1.43\times10^{+2}$	$2.97\times10^{+2}$	1.84	4.90×10^{-1}	4.85×10^{-2}
F_6	Ave	3.15×10^{-1}	9.23×10^{-9}	1.08×10^{-16}	4.88	3.15×10^{-1}	8.18×10^{-17}	6.58×10^{-1}	7.01×10^{-19}	3.15×10^{-26}
	std	9.98×10^{-2}	1.78×10^{-8}	4.00×10^{-17}	9.75×10^{-1}	9.98×10^{-2}	1.70×10^{-18}	3.38×10^{-1}	4.39×10^{-20}	6.31×10^{-28}
F_7	Ave	6.79×10^{-4}	6.92×10^{-2}	7.68×10^{-1}	3.88×10^{-2}	2.02×10^{-2}	5.37×10^{-1}	7.80×10^{-4}	2.71×10^{-5}	2.16×10^{-9}
	std	3.29×10^{-3}	2.87×10^{-2}	2.77	5.79×10^{-2}	7.43×10^{-3}	1.89×10^{-1}	3.85×10^{-4}	9.26×10^{-6}	1.24×10^{-7}

Received: March 24, 2020. Revised: April 27, 2020.

Table 2. Results for HOGO and other algorithms considering Multimodal test functions.

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	HOGO
F_8	Ave	$-5.11\times10^{+2}$	$-5.01\times10^{+2}$	$-2.75\times10^{+2}$	$-3.81\times10^{+2}$	$-6.92\times10^{+2}$	$-4.69\times10^{+1}$	$-6.14\times10^{+2}$	$-8.76\times10^{+2}$	$-1.2\times10^{+4}$
	std	$4.37\times10^{+1}$	$4.28\times10^{+1}$	$5.72\times10^{+1}$	$2.83\times10^{+1}$	$9.19\times10\times10^{+1}$	$3.94\times10^{+1}$	$9.32\times10^{+1}$	$5.92\times10^{+1}$	8.72×10^{-12}
F_9	Ave	1.23×10^{-1}	1.20×10^{-1}	$3.35\times10^{+1}$	$2.23\times10^{+1}$	$1.01\times10^{+2}$	4.85×10^{-2}	4.34×10^{-1}	6.90×10^{-1}	5.62×10^{-4}
	std	$4.11\times10^{+1}$	$4.01\times10^{+1}$	$1.19\times10^{+1}$	$3.25\times10^{+1}$	$1.89\times10^{+1}$	$3.91\times10^{+1}$	1.66	4.81×10^{-1}	3.21×10^{-2}
F_{10}	Ave	5.31×10^{-11}	5.20×10^{-11}	8.25×10^{-9}	$1.55\times10^{+1}$	1.15	2.83×10^{-8}	1.63×10^{-14}	8.03×10^{-16}	2.61×10^{-20}
	std	1.11×10^{-10}	1.08×10^{-10}	1.90×10^{-9}	8.11	7.87×10^{-1}	4.34×10^{-7}	3.14×10^{-15}	2.74×10^{-14}	2.14×10^{-18}
F_{11}	Ave	3.31×10^{-6}	3.24×10^{-6}	8.19	3.01×10^{-1}	5.74×10^{-1}	2.49×10^{-5}	2.29×10^{-3}	4.20×10^{-5}	1.56×10^{-10}
	std	4.23×10^{-5}	4.11×10^{-5}	3.70	2.89×10^{-1}	1.12×10^{-1}	1.34×10^{-4}	5.24×10^{-3}	4.73×10^{-4}	4.15×10^{-7}
F_{12}	Ave	9.16×10^{-8}	8.93×10^{-8}	2.65×10^{-1}	$5.21\times10^{+1}$	1.27	1.34×10^{-5}	3.93×10^{-2}	5.09×10^{-3}	4.87×10^{-5}
	std	4.88×10^{-7}	4.77×10^{-7}	3.14×10^{-1}	$2.47\times10^{+2}$	1.02	6.23×10^{-4}	2.42×10^{-2}	3.75×10^{-3}	3.96×10^{-4}
F_{13}	Ave	6.39×10^{-2}	6.26×10^{-2}	5.73×10^{-32}	$2.81\times10^{+2}$	6.60×10^{-2}	9.94×10^{-8}	4.75×10^{-1}	1.25×10^{-8}	0.00
	std	4.49×10^{-2}	4.39×10^{-2}	8.95×10^{-32}	$8.63\times10^{+2}$	4.33×10^{-2}	2.61×10^{-7}	2.38×10^{-1}	2.61×10^{-7}	0.00

Table 3. Results for HOGO and other algorithms considering Multimodal test functions with low dimension.

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	HOGO
F_{14}	Ave	4.39	2.77	3.61	6.79	$9.98\times10^{+1}$	1.26	3.71	1.08	9.91×10^{-1}
	std	4.41×10^{-2}	2.32	2.96	1.12	9.14×10^{-1}	6.86×10^{-1}	3.86	4.11×10^{-2}	6.52×10^{-12}
F_{15}	Ave	7.36×10^{-2}	9.09×10^{-3}	6.84×10^{-2}	5.15×10^{-2}	7.15×10^{-2}	1.01×10^{-2}	3.66×10^{-2}	8.21×10^{-3}	2.35×10^{-4}
	std	2.39×10^{-3}	2.38×10^{-3}	7.37×10^{-2}	3.45×10^{-3}	1.26×10^{-1}	3.75×10^{-3}	7.60×10^{-2}	4.09×10^{-3}	1.13×10^{-5}
F_{16}	Ave	-1.02	-1.02	-1.02	-1.01	-1.02	-1.02	-1.02	-1.02	-1.03
	std	4.19×10^{-7}	0.00	0.00	3.64×10^{-8}	4.74×10^{-8}	3.23×10^{-5}	7.02×10^{-9}	9.80×10^{-7}	4.52×10^{-10}
F_{17}	Ave	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	std	3.71×10^{-17}	9.03×10^{-16}	1.13×10^{-16}	9.45×10^{-15}	1.15×10^{-7}	7.61×10^{-4}	7.00×10^{-7}	5.39×10^{-5}	3.25×10^{-21}
F_{18}	Ave	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
	std	6.33×10^{-7}	6.59×10^{-5}	3.24×10^{-2}	1.94×10^{-10}	$1.48\times10^{+1}$	2.25×10^{-5}	7.16×10^{-6}	1.15×10^{-8}	5.32×10^{-19}
F_{19}	Ave	-3.81	-3.80	-3.86	-3.73	-3.77	-3.75	-3.84	-3.86	-3.86
	std	4.37×10^{-10}	3.37×10^{-15}	4.15×10^{-1}	9.69×10^{-4}	3.53×10^{-7}	2.55×10^{-3}	1.57×10^{-3}	6.50×10^{-7}	8.67×10^{-11}
F_{20}	Ave	-2.39	-3.32	-1.47	-2.17	-3.23	-2.84	-3.27	-2.81	-3.31
	std	4.37×10^{-1}	2.66×10^{-1}	5.32×10^{-1}	1.64×10^{-1}	5.37×10^{-2}	3.71×10^{-1}	7.27×10^{-2}	7.11×10^{-1}	3.51×10^{-5}
F_{21}	Ave	-5.19	-7.54	-4.57	-7.33	-7.38	-2.28	-9.65	-8.07	-10.15
	std	2.34	2.77	1.30	1.29	2.91	1.80	1.54	2.29	2.32×10^{-3}
F_{22}	Ave	-2.97	-8.55	-6.58	-1.00	-8.50	-3.99	-1.04	-10.01	-10.40
	std	1.37×10^{-2}	3.08	2.64	2.89×10^{-4}	3.02	1.99	2.73×10^{-4}	3.97×10^{-2}	4.52×10^{-8}
F_{23}	Ave	-3.10	-9.19	-9.37	-2.46	-8.41	-4.49	$-1.05\times10^{+1}$	-3.41	-10.55
	std	2.37	2.52	2.75	1.19	3.13	1.96	1.81×10^{-4}	1.11×10^{-2}	4.62×10^{-6}

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, M. Dehghani, Z. Montazeri, and J.M. Guerrero.; methodology, M. Dehghani and A. Dehghani.; software, M. Dehghani and A. Dehghani.; validation, J.M. Guerrero, K. Al-Haddad and O.P. Malik.; formal analysis, O.P. Malik and S. Saremi.; investigation, M. Dehgani and A. Dehghani.; resources, J.M. Guerrero.; data curation, S. Saremi and K. Al-Haddad; writing—original draft preparation, M. Dehghani and Z. Montazeri.; writing—review and editing, S. Saremi, O.P. Malik, and K. Al-Haddad.; visualization, M. Dehghani.; supervision, M. Dehghani.; project administration, M. Dehghani and Z. Montazeri; funding acquisition, J.M. Guerrero.

Acknowledgments

This work was supported by VILLUM FONDEN under the VILLUM Investigator Grant (no. 25920): Center for Research on Microgrids (CROM).

References

[1] 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", *Електротехніка і Електромеханіка*, No. 1, pp. 61-64, 2020.

[2] 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. 539-543, 2017.

[3] 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. 215-219, 2017.

[4] E. Stripling, S. vanden Broucke, K. Antonio, B. Baesens, and M. Snoeck, "Profit maximizing logistic model for customer churn prediction using genetic algorithms", *Swarm and Evolutionary Computation,* Vol. 40, pp. 116-130, 2018.

[5] I. V. Antonov, E. Mazurov, M. Borodovsky, and Y. A. Medvedeva, "Prediction of lncRNAs and their interactions with nucleic acids: benchmarking bioinformatics tools", *Briefings in Bioinformatics*, 2018.

[6] Y. Djenouri, A. Belhadi, and R. Belkebir, "Bees swarm optimization guided by data mining techniques for document information retrieval", *Expert Systems With Applications*, Vol. 94, pp. 126-136, 2018.

[7] N. Artrith, A. Urban, and G. Ceder, "Constructing first-principles phase diagrams of amorphous Li x Si using machine-learning-assisted sampling with an evolutionary algorithm", *The Journal of Chemical Physics*, Vol. 148, p. 241711, 2018.

[8] M. Dehghani, M. Mardaneh, Z. Montazeri, A. Ehsanifar, M. Ebadi, and O. Grechko, "Spring Search Algorithm for Simultaneous Placement of Distributed Generation and Capacitors", *Електротехніка і Електромеханіка*, No. 6, pp. 68-73, 2018.

[9] 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, 2020.

[10] Z. Montazeri and T. Niknam, "Optimal Utilization of Electrical Energy from Power Plants Based on Final Energy Consumption Using Gravitational Search Algorithm", *Електротехніка і Електромеханіка*, No. 4, pp. 70-73, 2018.

[11] M. Dehghani, Z. Montazeri, and O. P. Malik, "Energy commitment: a Planning of Energy Carrier Based on Energy Consumption", *Електротехніка і Електромеханіка*, No. 4, pp. 69-72, 2019.

[12] 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,* Vol. 5, pp. 62-71, 2018.

[13] 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 Conference on Electrical Engineering (ICEE),* Tehran, Iran, pp. 1372-1377, 2017.

[14] 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*

Received: March 24, 2020. Revised: April 27, 2020.

Engineering and Innovation (KBEI), Tehran, Iran, pp. 220-224, 2017.

[15] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation", In: *Proc. of Irish Conference on Artificial Intelligence and Cognitive Science*, pp. 103-110, 2002.

[16] P. Tarasewich and P. R. McMullen, "Swarm intelligence: power in numbers", *Communications of the ACM*, Vol. 45, pp. 62-67, 2002.

[17] T. Kohonen, *Self-organization and associative memory*, Vol. 8: Springer Science & Business Media, 2012.

[18] 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.

[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671-680, 1983.

[20] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search", *Simulation*, Vol. 76, pp. 60-68, 2001.

[21] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning", *Physica D: Nonlinear Phenomena*, Vol. 22, pp. 187-204, 1986.

[22] S. Mirjalili, "Ant Colony Optimisation", *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 33-42, 2019.

[23] J. C. Bansal, "Particle Swarm Optimization", *Evolutionary and Swarm Intelligence Algorithms*, ed: Springer, pp. 11-23, 2019.

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

[25] D. G. Yoo, J. H. Kim, and Z. W. Geem, "Overview of Harmony Search algorithm and its applications in Civil Engineering", *Evolutionary Intelligence*, Vol. 7, pp. 3-16, 2014.

[26] K.-L. Du and M. Swamy, "Particle swarm optimization", *Search and optimization by metaheuristics*, ed: Springer, pp. 153-173, 2016.

[27] 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. 210-214, 2017.

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

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

[30] 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.

[31] 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.

[32] A. Kaveh and M. Khayatazad, "A New Meta-Heuristic Method: Ray Optimization", *Computers & Structures*, Vol. 112, pp. 283-294, 2012.

[33] B. Alatas, "ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization", *Expert Systems with Applications*, Vol. 38, pp. 13170-13180, 2011.

[34] H. Du, X. Wu, and J. Zhuang, "Small-World Optimization Algorithm for Function Optimization", In: *Proc. of International Conference on Natural Computation*, Berlin, Heidelberg, pp. 264-273, 2006.

[35] A. Hatamlou, "Black Hole: A New Heuristic Optimization Approach for Data Clustering", *Information sciences*, Vol. 222, pp. 175-184, 2013.

[36] N. E. Karkalos, A. P. Markopoulos, and J. P. Davim, "Evolutionary-Based Methods", *Computational Methods for Application in Industry 4.0*, ed: Springer, pp. 11-31, 2019.

[37] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art", *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 4-31, 2011.

[38] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies–A Comprehensive Introduction", *Natural Computing*, Vol. 1, pp. 3-52, 2002.

[39] J. R. Koza, *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*: Stanford University, Department of Computer Science, 1990.

[40] S. Mirjalili, "Biogeography-Based Optimisation", *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 57-72, 2019.

[41] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances", *Handbook of metaheuristics*, ed: Springer, pp. 311-351, 2019.

[42] 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.

[43] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", In: *Proc. of Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, ed: Springer, pp. 65-74, 2010.

[44] 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.

[45] 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.

[46] G. Dhiman and V. Kumar, "Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems", *Knowledge-Based Systems*, Vol. 159, pp. 20-50, 2018.

[47] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[48] 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, pp. 57-64, 2020.

[49] 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, pp. 99-112, 2019.

[50] 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*, Vol. 9, pp. 5306-5310, 2019.

[51] M. Dehghani, Z. Montazeri, A. Dehghani, and O. P. Malik, "GO: Group Optimization", *Gazi University Journal of Science*, Vol. 33, 2020.

[52] M. Dehghani, Z. Montazeri, and O. P. Malik, "DGO: Dice Game Optimizer", *Gazi University Journal of Science*, Vol. 32, pp. 871-882, 2019.

[53] M. Dehghani, M. Mardaneh, O. P. Malik, and S. M. NouraeiPour, "DTO: Donkey Theorem Optimization", In: *Proc. of Iranian Conference on Electrical Engineering (ICEE)*, Yazd, Iran, pp.1855-1859, 2019.

[54] D. Mohammad, M. Zeinab, 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.

[55] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and Application", *Advances in Engineering Software*, Vol. 105, pp. 30-47, 2017.

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

[57] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, Vol.3, No.2, pp. 82-102, 1999.

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

[59] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems", *Computer-Aided Design*, Vol. 43, pp. 303-315, 2011.

[60] A. E. Eiben and C. A. Schippers, "On evolutionary exploration and exploitation", *Fundamenta Informaticae*, Vol. 35, pp. 35-50, 1998.