Received: October 20, 2021. Revised: October 30, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

POA: Puzzle Optimization Algorithm

Fatemeh Ahmadi Zeidabadi[1] **Mohammad Dehghani**[2]*

[1] *Department of Mathematics and Computer Sciences, Sirjan University of Technology, Sirjan, Iran*
[2]*Graduate of the Dept. of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran*
* Corresponding author's Email: adanbax@gmail.com

Abstract: Optimization plays a key role in various disciplines of science in order to achieve the optimal solution among all available solutions. Innovation and contribution of this paper is in introducing a new optimization algorithm called Puzzle Optimization Algorithm (POA) to solve various optimization problems. The main idea in the design of the proposed POA is the mathematical simulation of the process of solving a puzzle as an evolutionary optimizer. The various steps of POA are explained and then its mathematical model is presented. The main advantage and feature of the proposed POA is that it has no control parameters and therefore does not require parameter setting. The performance of the proposed POA is tested on twenty-three different objective functions. Also, in order to analyse the performance of POA in optimization, its results are compared with eight other optimization algorithms. The optimization results indicate the high and acceptable ability of the proposed POA to solve optimization problems. In addition, the simulation results show that the proposed POA is far better and much more competitive than the eight compared optimization algorithms.

Keywords: Optimization, Population-based, Optimization algorithm, Optimization problems, Puzzle, Puzzle Optimization.

1. Introduction

Optimization is a science in which an optimal solution for an optimization problem is presented among the available solutions according to the constraints of the problem [1]. From a mathematical point of view, an optimization problem consists of three main parts: a) variables, b) problem constraints, and c) objective function [2]. This means that in the optimization process, the problem variables must be adjusted in such a way according to all the constraints of the problem, that the objective function be optimized [3]. Optimization problem solving methods include two groups of deterministic methods and random methods. In deterministic algorithms, there is at most one way to continue the algorithm, which is negated, the algorithm is terminated and the global optimal value is introduced. In fact, none of the steps in deterministic algorithms uses random numbers or relationships, and the relationship between the steps is well defined and mathematically definable. Obtaining the optimal value using derivation is one of the simplest examples of deterministic algorithms. random algorithms may represent the quasi-optimal value, but this does not mean that their result is incorrect. In fact, random algorithms are used in situations where deterministic algorithms are not applicable to solve the problem or is very time consuming. In this situation, finding a quasi-optimal solution is definitely better than not getting any solution [4].

Random population-based optimization algorithms using random operators as well as random scanning of search space without using gradient information, are able to provide appropriate and acceptable solutions to optimization problems [5]. Population-based optimization algorithms first suggest a number of feasible solutions to the optimization problem. These proposed solutions are the same members of the algorithm population. Then, during successive iterations, based on a random scan of the search space and the steps of the algorithm, the proposed solutions are improved. After the full implementation of the algorithm on the optimization problem, a suitable solution to the problem is presented. The solution offered by the optimization algorithms even if is not exactly the same as the global optimal solution, it is close to global optimal solution [6]. Therefore, in comparing the performance of optimization algorithms, it is a better algorithm that provides a better quasi-optimal solution. For this reason, and with the aim of achieving better quasi-optimal solutions, researchers have developed numerous optimization algorithms. In this regard, optimization algorithms have been applied by scientists in various fields such as energy [7, 8], Energy Commitment (EC) [9, 10], protection [11], energy carriers [12, 13], and electrical engineering [14-18] to achieve the optimal solution.

The contribution and innovation of this paper is the design of a new optimization algorithm called Puzzle Optimization Algorithm (POA) that can be used to solve various optimization problems. The novelty of the proposed method is the use of relationships and cooperation of players in completing the puzzle, which has not been used so far to design optimization algorithms. The process of completing the puzzle is similar to an optimization process that ends over time by picking up the puzzle pieces to the main puzzle pattern, which is the optimal point. If this process is well modeled, it can generate the design of a powerful optimization algorithm. In the proposed POA, the main idea is to mathematically model the puzzle solving steps as an optimizer. The key idea in modeling the proposed method is to simulate the puzzle completion process in two steps. In the first step, each player tries to complete her/his own puzzle by imitating other players. Then, in the second stage, the players suggest the appropriate pieces to complete the puzzle to the player who has not been able to complete his puzzle well. The mathematical model of POA is presented and its performance is tested on a set of twenty-three standard objective functions. The performance results of the proposed POA are compared with eight other well-known optimization algorithms.

The rest of the paper is organized in such a way that in Section 2, the optimization algorithms are reviewed. The various steps and mathematical models of the proposed POA are presented in Section 3. Simulation studies are presented in Section 4. Finally, in Section 5, conclusions and suggestions for future studies are presented.

2. Background

Optimization algorithms have been developed inspired by various physical phenomena, living behaviors, genetic sciences, various individual and group games, and any other process that has an evolutionary process. Hence, in general, optimization algorithms can be grouped into four groups a) swarm-based, b) evolutionary-based, c) physics-based, and d) game-based optimization algorithms based on the main idea.

Swarm-based optimization algorithms are designed based on simulation of the swarming behaviors of living organisms, animals, plants, and insects in nature. Particle Swarm Optimization (PSO) is one of the most popular and widely used algorithms which is developed based on simulation of fishes and birds swarming behavior [19]. Simulation of ant swarm motion in finding the shortest path between the nest and the food source, is applied in the design of the Ant Colony Optimization (ACO) algorithm [20]. Simulation of leadership hierarchy in the gray wolf herd and the behavior of gray wolves during hunting has been used in the design of Gray Wolf Optimizer (GWO) [21]. Marine Predators Algorithm (MPA) is another swarm-based optimization algorithm that is designed according to the movement strategies that marine predators use when trapping their prey in the oceans [22]. Some other algorithms in this group are: Doctor and Patients Optimization (DPO) [10], Teaching-Learning-Based Optimization (TLBO) [23], Whale Optimization Algorithm (WOA) [24], Two Stage Algorithm(TSO) [25], Donkey Theorem Optimization (DTO) [26], Group Mean Based Optimizer (GMBO) [27], Cat and Mouse-Based Optimizer (CMBO) [28], Following Optimization Algorithm (FOA) [29], Tunicate Swarm Algorithm (TSA) [30], Good and Bad Groups Based Optimizer (GBGBO) [31], Rat Swarm Optimizer (RSO) [32], Good and Bad and Ugly Optimizer (GBUO) [33], and Seagull Optimization Algorithm (SOA) [34].

Evolutionary-based optimization algorithms are introduced based on modeling the genetic science, reproductive process, and evolutionary laws. Genetic Algorithm (GA), which is one of the most prominent optimization algorithms, belongs to this group. The design of GA is inspired by reproductive process and Darwin's theory of evolution, while it is simulated using three operators: selection, crossover, and mutation [35]. Artificial Immune System (AIS) algorithm is among the algorithm inspired by the mechanism of the human body that fall into the category of evolutionary-based optimization algorithms [36]. Some other algorithms in this group are: Artificial Infectious Disease (AID) [37], Evolutionary Programming (EP) [38] , Cultural Algorithm [39], Genetic Programming (GP) [40], Evolution Strategy (ES) [41], Biogeography-Based

Optimizer (BBO) [42] and Differential Evolution (DE) [43].

Physics-based optimization algorithms are developed inspired by various physical phenomena and laws. Simulation of metal melting and cooling process has been used in designing the Simulated Annealing (SA) [44] algorithm. Gravitational force modeling that objects apply to each other at different distances is used in the design of Gravitational Search Algorithm (GSA) [45]. Momentum law simulation, while applying the laws of motion in physics, has been used in the design of the Momentum Search Algorithm (MSA) [46]. The simulation of the Hooke law in a system consisting of connected weights and springs that exert force on each other has been used in the design of Spring Search Algorithm (SSA) [47]. Some other algorithms in this group are: Central Force Optimization (CFO) [48], Charged System Search (CSS) [49], Flow Direction Algorithm (FDA) [50], Henry Gas Solubility Optimization (HGSO) [51], and Binary Spring Search Algorithm (BSSA) [52].

Game-based optimization algorithms are developed based on mathematical modeling of rules and behavior of players in different individual or group games. Volleyball Premier League (VPL) is developed based on the competition and interaction among volleyball teams and the coaching process during a volleyball match [53]. Football Game Based Optimization (FGBO) algorithm is designed based on modeling the interactions between football clubs and simulating four stages: holding matches, transferring players, training, promotion and relegation of clubs in a season [54]. Darts Game Optimizer (DGO) is introduced based on modeling the behavior of players in darts match during throwing darts and collecting points [55]. Some other algorithms in this group are: Hide Object Game Optimizer (HOGO) [56], Ring Game Based Optimizer (RTGBO) [57], Orientation Search Algorithm (OSA) [58], Tug of War Optimization (TWO) [59], Dice Game Optimization (DGO) [15], and Shel Game Optimization (SGO) [60].

3. Puzzle optimization algorithm

In this section, the proposed Puzzle Optimization Algorithm (POA) is introduced and its mathematical modeling is presented for use in solving optimization problems.

The proposed POA is a population-based algorithm that has been developed based on puzzle game simulation. Therefore, POA belongs to the group of game-based optimization algorithms. Each member of the population is considered as a puzzle in such a way that the pieces of the puzzle determine the variables of the problem. The better the pieces of the puzzle are put in place, the more points can be considered for solving that puzzle, which can be modeled using the evaluation and the value of the objective function. In the proposed POA, in solving each puzzle, guidance is taken from other members of the population, especially the best member of the population.

In the proposed POA, which is a population-based algorithm, each population member is a feasible solution to the optimization problem. In fact, each member of the population determines the values of the problem variables. Thus, the algorithm population in POA can be mathematically modeled using a matrix which is specified in Eq. (1).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} \cdots x_{1,d} \cdots x_{1,m} \\ \vdots \quad \ddots \quad \vdots \quad \iddots \quad \vdots \\ x_{i,1} \cdots x_{i,d} \cdots x_{i,m} \\ \vdots \quad \iddots \quad \vdots \quad \ddots \quad \vdots \\ x_{N,1} \cdots x_{N,d} \cdots x_{N,m} \end{bmatrix}_{N\times m} \tag{1}$$

Here, X is the population of puzzles, X_i is the i'th puzzle, N is the number of population of puzzles, m is the number of problem variables, and $x_{i,d}$ is the value of d'th variables suggested by i'th puzzle.

Given that each member of the population is a proposed solution to the optimization problem, the value of the objective function can be evaluated. Therefore, equal to the number of population members, the objective function is evaluated, which the obtained values for objective function are simulated using a vector in Eq. (2).

$$F = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{2}$$

Here, F is the vector of obtained values for objective function and f_i is the value of objective function of i'th puzzle.

Based on the comparison of the values obtained for the objective function, the member that provides the best value for the objective function is recognized as the best member of the population. The best member can be determined using Eq. (3).

$$B = X_k, f_k = \min(F) \tag{3}$$

Here, B is the best member and X_k is the k'th puzzle with minimum of objective function equal to

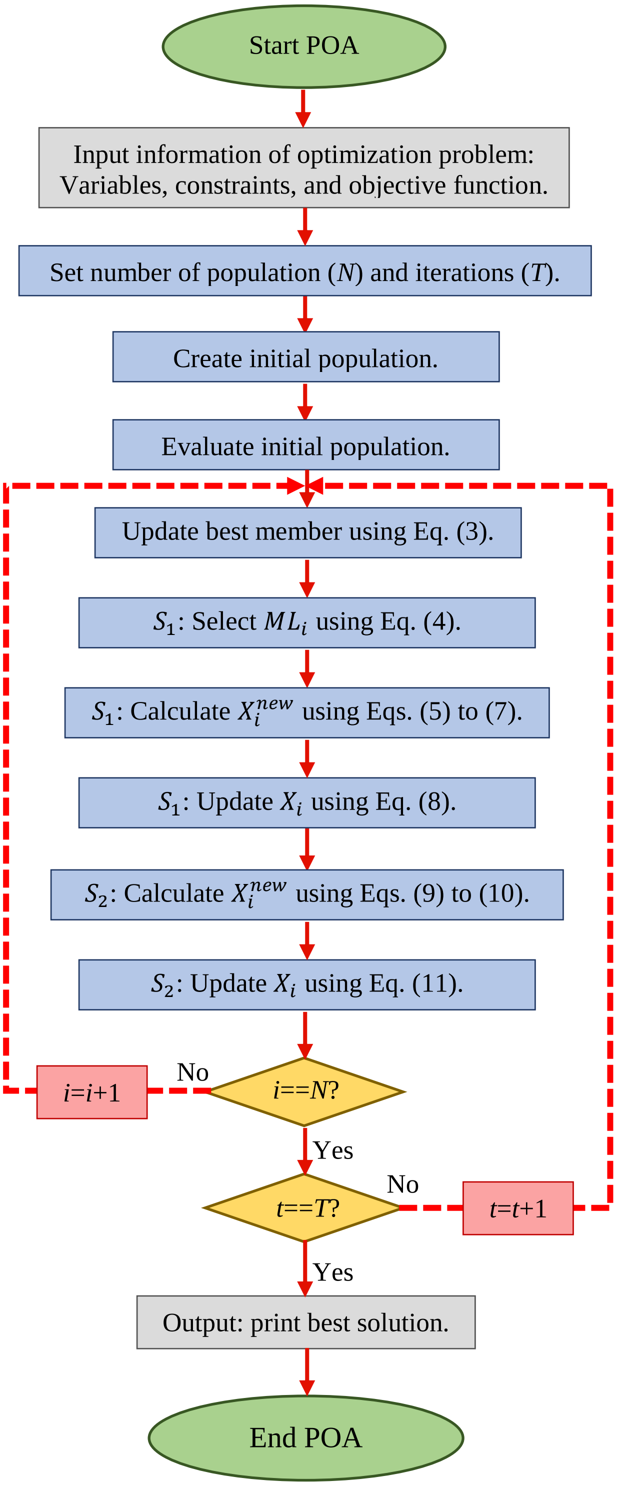
Start POA
Input information of optimization problem: Variables, constraints, and objective function.
Set number of population (N) and iterations (T).
Create initial population.
Evaluate initial population.
Update best member using Eq. (3).
S_1: Select ML_i using Eq. (4).
S_1: Calculate X_i^{new} using Eqs. (5) to (7).
S_1: Update X_i using Eq. (8).
S_2: Calculate X_i^{new} using Eqs. (9) to (10).
S_2: Update X_i using Eq. (11).
i=i+1
No
i==N?
Yes
t==T?
No
t=t+1
Yes
Output: print best solution.
End POA

Figure. 1 Flowchart of POA

f_k.

In the proposed POA, population members are updated in two stages. In the first stage, each member of the population is updated based on the guidance of the other members. In the second stage, each member of the population tries to complete its puzzle based on the use of puzzle pieces suggested by other members.

The concept expressed in the first stage is mathematically modeled using Eqs. (4) to (8).

$$GM_i = X_g, \qquad g \in \{1,2,3,\ldots,N\} \tag{4}$$

$$dx_{i,d} = \begin{cases} (GM_{i,d} - I \times x_{i,d}), & F_g < F_i \\ (x_{i,d} - I \times GM_{i,d}), & else \end{cases} \tag{5}$$

$$I = round\ (1 + rand) \tag{6}$$

$$X_i^{new} = X_i + r \times dX_i \tag{7}$$

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & else \end{cases} \tag{8}$$

Here, GM_i is the guiding member of i'th puzzle, $GM_{i,d}$ is its d'th dimension, F_g is its value of objective function, I is a random number that can be 1 or 2, $dx_{i,d}$ is the changes of d'th dimension of i'th puzzle, r is a random number in [0 1] interval, X_i^{new} is the new status of i'th puzzle, and F_i^{new} is its value of objective function.

In the second stage, each member of the population updates its status using puzzle pieces suggested by other members of the population. This process is mathematically modeled using Eqs. (9) to (11).

$$N_P = round\left(0.5 \times \left(1 - \frac{t}{T}\right) \times N\right), \tag{9}$$

$$x_{i,d_j}^{new} = x_{h,d_j}, \qquad \begin{cases} h \in \{1,2,3,\ldots,N\} \\ j \in \{1,2,3,\ldots,N_p\} \\ d_j \in \{1,2,3,\ldots,m\} \end{cases} \tag{10}$$

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & else \end{cases} \tag{11}$$

Here, N_P is the number of suggested puzzle pieces, t is the iteration counter, T is the maximum number of iterations, x_{i,d_j}^{new} is the new value for d_j'th dimension of i'th puzzle, and x_{h,d_j} is the selected suggested puzzle piece from h'th puzzle which h is selected randomly.

After updating all members of the population according to the first and second stages, an iteration of the algorithm is performed and the new status of the members of the population is determined. Given that the proposed POA is an iteration-based algorithm, the algorithm is updated according to Eqs. (3) to (11) until the last iteration. After completing the iterations of the algorithm, POA presents the best quasi-optimal solution to the optimization problem. The various steps of POA using the flowchart are shown in Fig. 1.

Received: October 20, 2021. Revised: October 30, 2021.

Table 1. Results of POA and other algorithms for unimodal test functions

		POA	MPA [22]	GSA [45]	HOGO [56]	GWO [21]	PSO [19]	WOA [24]	TLBO [23]	GA [35]
F_1	ave	1.72×10^{-64}	3.2715×10^{-21}	2.81×10^{-1}	4.36×10^{-36}	7.86×10^{-10}	4.98×10^{-9}	2.1741×10^{-9}	3.55×10^{-2}	1.95×10^{-12}
	std	7.18×10^{-73}	4.6153×10^{-21}	1.11×10^{-1}	6.84×10^{-37}	8.11×10^{-9}	1.40×10^{-8}	7.3985×10^{-25}	1.06×10^{-1}	2.01×10^{-11}
F_2	ave	2.74×10^{-68}	1.57×10^{-12}	3.96×10^{-1}	5.89×10^{-49}	5.99×10^{-20}	7.29×10^{-4}	0.5462	3.23×10^{-5}	6.53×10^{-18}
	std	5.46×10^{-82}	1.42×10^{-12}	1.41×10^{-1}	3.25×10^{-46}	1.11×10^{-17}	1.84×10^{-3}	1.7377×10^{-16}	8.57×10^{-5}	5.10×10^{-17}
F_3	ave	8.66×10^{-43}	0.0864	$4.31\times10^{+1}$	6.85×10^{-26}	9.19×10^{-5}	$1.40\times10^{+1}$	1.7634×10^{-8}	$4.91\times10^{+3}$	7.70×10^{-10}
	std	1.68×10^{-48}	0.1444	8.97	4.52×10^{-29}	6.16×10^{-4}	7.13	1.0357×10^{-23}	$3.89\times10^{+3}$	7.36×10^{-9}
F_4	ave	2.93×10^{-40}	2.6×10^{-8}	8.80×10^{-1}	2.15×10^{-27}	8.73×10^{-1}	6.00×10^{-1}	2.9009×10^{-5}	$1.87\times10^{+1}$	$9.17\times10^{+1}$
	std	8.62×10^{-43}	9.25×10^{-9}	2.50×10^{-1}	5.68×10^{-30}	1.19×10^{-1}	1.72×10^{-1}	1.2121×10^{-20}	8.21	$5.67\times10^{+1}$
F_5	ave	1.99×10^{-1}	46.049	$1.18\times10^{+2}$	3.42×10^{-1}	$8.91\times10^{+2}$	$4.93\times10^{+1}$	41.7767	$7.37\times10^{+2}$	$5.57\times10^{+2}$
	std	5.89×10^{-7}	0.4219	$1.43\times10^{+2}$	3.84×10^{-2}	$2.97\times10^{+2}$	$3.89\times10^{+1}$	2.5421×10^{-3}	$1.98\times10^{+3}$	$4.16\times10^{+1}$
F_6	ave	0	0.398	3.15×10^{-1}	2.69×10^{-26}	8.18×10^{-17}	9.23×10^{-9}	1.6085×10^{-9}	4.88	3.15×10^{-1}
	std	0	0.1914	9.98×10^{-2}	5.62×10^{-28}	1.70×10^{-18}	1.78×10^{-8}	4.6240×10^{-25}	9.75×10^{-1}	9.98×10^{-2}
F_7	ave	6.91×10^{-15}	0.0018	2.02×10^{-2}	1.85×10^{-9}	5.37×10^{-1}	6.92×10^{-2}	0.0205	3.88×10^{-2}	6.79×10^{-4}
	std	2.15×10^{-14}	0.001	7.43×10^{-3}	1.05×10^{-7}	1.89×10^{-1}	2.87×10^{-2}	1.5515×10^{-18}	5.79×10^{-2}	3.29×10^{-3}

4. Simulation study

In this section, simulation studies and performance analysis of the proposed POA in solving optimization problems are presented. A set of twenty-three standard objective functions of various types of fixed-dimensional multimodal, high-dimensional multimodal, and unimodal are used as benchmark functions. Also, in order to analyse the optimization results obtained from POA, these results are compared with the performance of eight well-known optimization algorithms including: GSA [45], PSO [19], TLBO [23], GA [35], MPA [22], GWO [21], WOA [24], and HOGO [56]. The simulation results are reported using two indicators of the mean of the best solutions obtained (ave) and their standard deviation (std).

In the first case, the performance of the optimization algorithms on the objective functions of the unimodal type is evaluated. For this purpose, the proposed POA and eight competing algorithms have been implemented on the F1 to F7 objective functions. The optimization results of these objective functions are shown in Table 1. The results obtained from POA show the high ability of the proposed algorithm to solve optimization problems and provide quasi-optimal solutions. Also, comparing the simulation results of eight competing algorithms, shows that POA has been able to provide better results than similar algorithms and has better performance.

In the second case, the ability of the proposed POA and eight competing algorithms to solve the high-dimensional multimodal objective functions are tested. For this purpose, the mentioned algorithms have been implemented on the objective functions of F8 to F13 and the optimization results obtained from them are presented in Table 2. The results of optimization of F8 to F13 objective functions using POA show that the proposed algorithm is able to provide suitable and acceptable quasi-optimal solutions. The optimization results obtained from the eight competing algorithms show the superiority of the proposed POA in solving the high-dimensional multimodal objective functions over the eight algorithms.

In the third case, the analysis of the performance of optimization algorithms in solving fixed-dimensional multimodal optimization problems is discussed. In this regard, the ability of the proposed POA and eight competing algorithms to solve the objective functions of F14 to F23 are tested. The optimization results of these objective functions are presented in Table 3. Based on the results of this table, the proposed POA has been able to provide optimal solutions in solving such objective functions. Also,

Table 3. Results of POA and other algorithms for fixed-dimensional multimodal test functions

		POA	MPA [22]	GSA [45]	HOGO [56]	GWO [21]	PSO [19]	WOA [24]	TLBO [23]	GA [35]
F_{14}	ave	0.998	0.998	$9.98\times10^{+1}$	9.91×10^{-1}	1.26	2.77	0.998	6.79	4.39
	std	2.85×10^{-19}	4.27×10^{-14}	9.14×10^{-1}	5.25×10^{-12}	6.86×10^{-1}	2.32	9.4336×10^{-10}	1.12	4.41×10^{-2}
F_{15}	ave	5.31×10^{-6}	0.003	7.15×10^{-2}	1.25×10^{-4}	1.01×10^{-2}	9.09×10^{-3}	0.0049	5.15×10^{-2}	7.36×10^{-2}
	std	6.05×10^{-15}	4.09×10^{-11}	1.26×10^{-1}	2.15×10^{-5}	3.75×10^{-3}	2.38×10^{-3}	3.4910×10^{-8}	3.45×10^{-3}	2.39×10^{-3}
F_{16}	ave	-1.0316	−1.03	-1.02	-1.03	-1.02	-1.02	−1.03	-1.01	-1.02
	std	6.16×10^{-17}	4.4652×10^{-14}	4.74×10^{-8}	3.26×10^{-10}	3.23×10^{-5}	0.00	9.9301×10^{-6}	3.64×10^{-8}	4.19×10^{-7}
F_{17}	ave	3.98×10^{-1}	3.99×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	4.07×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	std	6.70×10^{-24}	9.1235×10^{-15}	1.15×10^{-7}	2.45×10^{-21}	7.61×10^{-4}	9.03×10^{-16}	2.4825×10^{-7}	9.45×10^{-15}	3.71×10^{-17}
F_{18}	ave	3	3	3	3	3	3	3	3	3
	std	3.57×10^{-24}	1.9584×10^{-15}	$1.48\times10^{+1}$	4.61×10^{-19}	2.25×10^{-5}	6.59×10^{-5}	5.6984×10^{-5}	1.94×10^{-10}	6.33×10^{-7}
F_{19}	ave	-3.8627	−3.86	-3.77	-3.86	-3.75	-3.80	−3.86	-3.73	-3.81
	std	9.19×10^{-17}	4.2428×10^{-14}	3.53×10^{-7}	6.23×10^{-11}	2.55×10^{-3}	3.37×10^{-15}	3.1916×10^{-5}	9.69×10^{-4}	4.37×10^{-10}
F_{20}	ave	-3.36	−3.32	-3.23	-3.31	-2.84	-3.32	−3.24	-2.17	-2.39
	std	2.08×10^{-8}	1.1421×10^{-6}	5.37×10^{-2}	2.95×10^{-5}	3.71×10^{-1}	2.66×10^{-1}	7.9441×10^{-6}	1.64×10^{-1}	4.37×10^{-1}
F_{21}	ave	-10.15	−10.15	-7.38	-10.15	-2.28	-7.54	−7.40	-7.33	-5.19
	std	2.16×10^{-14}	2.5361×10^{-9}	2.91	1.05×10^{-3}	1.80	2.77	2.3819×10^{-7}	1.29	2.34
F_{22}	ave	-10.40	−10.40	-8.50	-10.40	-3.99	-8.55	−8.81	-1.00	-2.97
	std	8.75×10^{-15}	2.8154×10^{-11}	3.02	3.25×10^{-8}	1.99	3.08	6.7524×10^{-10}	2.89×10^{-4}	1.37×10^{-2}
F_{23}	ave	-10.55	−10.53	-8.41	-10.55	-4.49	-9.19	−10.0003	-2.46	-3.10
	std	1.96×10^{-15}	3.9861×10^{-9}	3.13	2.91×10^{-6}	1.96	2.52	9.1357×10^{-7}	1.19	2.37

Table 2. Results of POA and other algorithms for high-dimensional multimodal test functions

		POA	MPA [22]	GSA [45]	HOGO [56]	GWO [21]	PSO [19]	WOA [24]	TLBO [23]	GA [35]
F_8	ave	$-1.2\times10^{+4}$	−3594.16321	$-6.92\times10^{+2}$	$-1.2\times10^{+4}$	$-4.69\times10^{+1}$	$-5.01\times10^{+2}$	−1663.9782	$-3.81\times10^{+2}$	$-5.11\times10^{+2}$
	std	2.61×10^{-17}	811.32651	$9.19\times10\times10^{+1}$	8.72×10^{-12}	$3.94\times10^{+1}$	$4.28\times10^{+1}$	716.3492	$2.83\times10^{+1}$	$4.37\times10^{+1}$
F_9	ave	0	140.1238	$1.01\times10^{+2}$	5.62×10^{-4}	4.85×10^{-2}	1.20×10^{-1}	4.2011	$2.23\times10^{+1}$	1.23×10^{-1}
	std	0	26.3124	$1.89\times10^{+1}$	3.21×10^{-2}	$3.91\times10^{+1}$	$4.01\times10^{+1}$	4.3692×10^{-15}	$3.25\times10^{+1}$	$4.11\times10^{+1}$
F_{10}	ave	2.59×10^{-25}	9.6987×10^{-12}	1.15	2.61×10^{-20}	2.83×10^{-8}	5.20×10^{-11}	0.3293	$1.55\times10^{+1}$	5.31×10^{-11}
	std	1.68×10^{-28}	6.1325×10^{-12}	7.87×10^{-1}	2.14×10^{-18}	4.34×10^{-7}	1.08×10^{-10}	1.9860×10^{-16}	8.11	1.11×10^{-10}
F_{11}	ave	0	0	5.74×10^{-1}	1.56×10^{-10}	2.49×10^{-5}	3.24×10^{-6}	0.1189	3.01×10^{-1}	3.31×10^{-6}
	std	0	0	1.12×10^{-1}	4.15×10^{-7}	1.34×10^{-4}	4.11×10^{-5}	8.9991×10^{-17}	2.89×10^{-1}	4.23×10^{-5}
F_{12}	ave	1.06×10^{-9}	0.0851	1.27	4.87×10^{-5}	1.34×10^{-5}	8.93×10^{-8}	1.7414	$5.21\times10^{+1}$	9.16×10^{-8}
	std	7.93×10^{-14}	0.0052	1.02	3.96×10^{-4}	6.23×10^{-4}	4.77×10^{-7}	8.1347×10^{-12}	$2.47\times10^{+2}$	4.88×10^{-7}
F_{13}	ave	0.00	0.4901	6.60×10^{-2}	0.00	9.94×10^{-8}	6.26×10^{-2}	0.3456	$2.81\times10^{+2}$	6.39×10^{-2}
	std	0.00	0.1932	4.33×10^{-2}	0.00	2.61×10^{-7}	4.39×10^{-2}	3.25391×10^{-12}	$8.63\times10^{+2}$	4.49×10^{-2}

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

References

[1] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, Z. Montazeri, P. Trojovský, and G. Dhiman, "AMBO: All Members-Based Optimizer for Solving Optimization Problems", *Computers, Materials \& Continua*, Vol. 70, No. 2, pp. 2905--2921, 2022.

[2] M. Dehghani, Z. Montazeri, A. Dehghani, R. R. Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, "MLO: Multi Leader Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 364-373, 2020.

[3] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, Z. Montazeri, P. Trojovský, and G. Dhiman, "MLA: A New Mutated Leader Algorithm for Solving Optimization Problems", *Computers, Materials \& Continua*, Vol. 70, No. 3, pp. 5631--5649, 2022.

[4] M. Dehghani, and P. Trojovský, "Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization", *Sensors*, Vol. 21, No. 13, p. 4567, 2021.

[5] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, "MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems," *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 472-479, 2021.

[6] M. Dehghani, Z. Montazeri, A. Dehghani, O. P. Malik, R. M. Menendez, G. Dhiman, N. Nouri, A. Ehsanifar, J. M. Guerrero, and R. A. R. Mendoza, "Binary spring search algorithm for solving various optimization problems", *Applied Sciences*, Vol. 11, No. 3, p. 1286, 2021.

[7] M. Dehghani, Z. Montazeri, and O. P. Malik, "ENERGY COMMITMENT: A PLANNING OF ENERGY CARRIER BASED ON ENERGY CONSUMPTION", *Electrical Engineering & Electromechanics*, No. 4, pp. 69-72, 2019.

[8] M. Dehghani, M. Mardaneh, O. P. Malik, J. M. Guerrero, C. Sotelo, D. Sotelo, M. N. Heris, K. A. Haddad, and R. A. R. Mendoza, "Genetic Algorithm for Energy Commitment in a Power System Supplied by Multiple Energy Carriers", *Sustainability*, Vol. 12, No. 23, p. 10053, 2020.

[9] M. Dehghani, M. Mardaneh, O. P. Malik, J. M. Guerrero, R. M. Menendez, R. A. R. Mendoza, J. Matas, and A. Abusorrah, "Energy Commitment for a Power System Supplied by Multiple Energy Carriers System using Following Optimization Algorithm", *Applied Sciences*, Vol. 10, No. 17, p. 5862, 2020.

[10] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, R. A. R. Mendoza, J. Matas, J. C. Vasquez, and L. P. Arroyo, "A new "Doctor and Patient" optimization algorithm: An application to energy commitment problem", *Applied Sciences*, Vol. 10, No. 17, p. 5791, 2020.

[11] 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.

[12] M. Dehghani, Z. Montazeri, A. Ehsanifar, A. R. Seifi, M. J. Ebadi, and O. M. Grechko, "Planning of energy carriers based on final energy consumption using dynamic programming and particle swarm optimization", *Electrical Engineering & Electromechanics*, No. 5, pp. 62-71, 2018.

[13] Z. Montazeri and T. Niknam, "Energy Carriers Management Based on Energy Consumption", In: *Proc. of IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp.0539-0543, 2017.

[14] M. Dehghani, Z. Montazeri, and O. 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, 2020.

[15] M. Dehghani, Z. Montazeri, O. P. Malik, K. A. Haddad, J. M. Guerrero, and G. Dhiman, "A New Methodology Called Dice Game Optimizer for Capacitor Placement in Distribution Systems", *Electrical Engineering & Electromechanics*, No. 1, pp. 61-64, 2020.

[16] 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 Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 0215-0219, 2017.

[17] Z. Montazeri, and T. Niknam, "Optimal utilization of electrical energy from power plants based on final energy consumption using gravitational search algorithm", *Electrical Engineering & Electromechanics*, No. 4, pp. 70-73, 2018.

[18] M. Dehghani, M. Mardaneh, Z. Montazeri, A. Ehsanifar, M. J. Ebadi, and O. M. Grechko, "Spring search algorithm for simultaneous placement of distributed generation and capacitors", *Electrical Engineering & Electromechanics*, No. 6, pp. 68-73, 2018.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, ed: IEEE, pp. 1942-1948, 1995.

[20] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization", *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28-39, 2006.

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

[22] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic", *Expert Systems with Applications*, Vol. 152, p. 113377, 2020.

[23] 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, No. 3, pp. 303-315, 2011.

[24] S. Mirjalili and A. Lewis, "The whale optimization algorithm", *Advances in Engineering Software*, Vol. 95, pp. 51-67, 2016.

[25] S. A. Doumari, H. Givi, M. Dehghani, Z. Montazeri, V. Leiva, and J. M. Guerrero, "A New Two-Stage Algorithm for Solving Optimization Problems", *Entropy*, Vol. 23, No. 4, p. 491, 2021.

[26] M. Dehghani, M. Mardaneh, O. P. Malik, and S. M. Nouraeipour, "DTO: Donkey Theorem Optimization", In: *Proc. of 27th Iranian Conference on Electrical Engineering (ICEE)*, Yazd, Iran, pp. 1855-1859, 2019.

[27] M. Dehghani, Z. Montazeri, and Š. Hubálovský, "GMBO: Group Mean-Based Optimizer for Solving Various Optimization Problems", *Mathematics*, Vol. 9, No. 11, p. 1190, 2021.

[28] M. Dehghani, Š. Hubálovský, and P. Trojovský, "Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm", *Sensors*, Vol. 21, No. 15, p. 5214, 2021.

[29] M. Dehghani, M. Mardaneh, and O. 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.

[30] 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.

[31] A. Sadeghi, S. A. Doumari, M. Dehghani, Z. Montazeri, P. Trojovský, and H. J. Ashtiani, "A New "Good and Bad Groups-Based Optimizer" for Solving Various Optimization Problems", *Applied Sciences*, Vol. 11, No. 10, p. 4382, 2021.

[32] 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*, pp. 1-26, 2020.

[33] H. Givi, M. Dehghani, Z. Montazeri, R. M. Menendez, R. A. R. Mendoza, and N. Nouri, "GBUO: "The Good, the Bad, and the Ugly" Optimizer", *Applied Sciences*, Vol. 11, No. 5, p. 2042, 2021.

[34] G. Dhiman, K. K. Singh, M. Soni, A. Nagar, M. Dehghani, A. Slowik, A. Kaur, A. Sharma, E. H. Houssein, and K. Cengiz, "MOSOA: A new multi-objective seagull optimization algorithm", *Expert Systems with Applications*, pp. 114150, 2020.

[35] A. Bose, T. Biswas, and P. Kuila, "A novel genetic algorithm based scheduling for multi-core systems", *Smart Innovations in Communication and Computational Sciences*, ed: Springer, p. 45-54, 2019.

[36] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system", *Evolutionary computation*, Vol. 8, No. 4, pp. 443-473, 2000.

[37] G. Huang, "Artificial infectious disease optimization: A SEIQR epidemic dynamic model-based function optimization algorithm", *Swarm and Evolutionary Computation*, Vol. 27, pp. 31-67, 2016.

[38] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through simulated evolution", 1966.

[39] R. G. Reynolds, "An introduction to cultural algorithms", In *Proceedings of the Third Annual*

Conference on Evolutionary Programming, Vol. 24, pp. 131-139, 1994.

[40] J. R. Koza, "Genetic programming as a means for programming computers by natural selection", *Statistics and Computing*, Vol. 4, No. 2, pp. 87-112, 1994.

[41] H. G. Beyer and H. P. Schwefel, "Evolution strategies–A comprehensive introduction", *Natural Computing*, Vol. 1, No. 1, pp. 3-52, 2002.

[42] D. Simon, "Biogeography-based optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp. 702-713, 2008.

[43] 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.

[44] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.

[45] E. Rashedi, H. N. Pour, and S. Saryazdi, "GSA: a gravitational search algorithm", *Information Sciences*, Vol. 179, No. 13, pp. 2232-2248, 2009.

[46] M. Dehghani and H. Samet, "Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law", *SN Applied Sciences*, Vol. 2, No. 10, pp. 1-15, 2020.

[47] M. Dehghani, Z. Montazeri, G. Dhiman, O. Malik, R. M. Menendez, R. A. R. Mendoza, A. Dehghani, J. M. Guerrero, and L. P. Arroyo, "A spring search algorithm applied to engineering optimization problems", *Applied Sciences*, Vol. 10, No. 18, p. 6173, 2020.

[48] R. A. Formato, "Central force optimization", *Prog Electromagn Res*, Vol. 77, pp. 425-491, 2007.

[49] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search", *Acta Mechanica*, Vol. 213, No. 3, pp. 267-289, 2010.

[50] H. Karami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems", *Computers & Industrial Engineering*, Vol. 156, p. 107224, 2021.

[51] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. A. Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm", *Future Generation Computer Systems*, Vol. 101, pp. 646-667, 2019.

[52] M. Dehghani, Z. Montazeri, A. Dehghani, N. Nouri, and A. Seifi, "BSSA: Binary Spring Search Algorithm", In: *Proc. of IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 0220-0224, 2017.

[53] R. Moghdani, and K. Salimifard, "Volleyball premier league algorithm", *Applied Soft Computing*, Vol. 64, pp. 161-185, 2018.

[54] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. Malik, and V. Kumar, "Football game based optimization: An application to solve energy commitment problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, pp. 514-523, 2020.

[55] 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, pp. 286-294, 2020.

[56] M. Dehghani, Z. Montazeri, S. Saremi, A. Dehghani, O. P. Malik, K. A. Haddad, and J. M. Guerrero, "HOGO: Hide Objects Game Optimization", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 216-225, 2020.

[57] S. A. Doumari, H. Givi, M. Dehghani, and O. P. Malik, "Ring Toss Game-Based Optimization Algorithm for Solving Various Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 545-554, 2021.

[58] 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.

[59] A. Kaveh, and A. Zolghadr, "A novel meta-heuristic algorithm: tug of war optimization", *Iran University of Science & Technology*, Vol. 6, No. 4, pp. 469-492, 2016.

[60] 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.