Received: February 13, 2020

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Shell Game Optimization: A Novel Game-Based Algorithm

Mohammad Dehghani[1]* Zeinab Montazeri[1] Om Parkash Malik[2]
Hadi Givi[3] Josep M. Guerrero[4]

[1]Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran
[2]Department of Electrical Engineering, University of Calgary, Calgary Alberta, Canada
[3]Department of Electrical Engineering, Faculty of Engineering, University of Shahreza, Shahreza, Iran
[4]Center for Research on Microgrids (CROM), Department of Energy Technology, Aalborg University, Aalborg, Denmark
* Corresponding author's Email: adanbax@gmail.com

Abstract: This article presents a new game-based optimization method entitled Shell Game Optimization (SGO). The novelty of this article is simulating the rules of a game known as shell game to design an algorithm for solving optimization problems in different fields of science. The key idea of the SGO is to find the ball hidden under one of the three shells, which should be guessed by players. The main feature and advantage of SGO is that it does not have any control parameters and hence, there is no need to set parameters. SGO is mathematically modeled and implemented on 23 well-known benchmark test functions as well as on a real life-engineering problem entitled pressure vessel design problem. Moreover, SGO is compared with eight optimization algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Teaching Learning Based Optimization (TLBO), Grey Wolf Optimizer (GWO), Grasshopper Optimization Algorithm (GOA), Spotted Hyena Optimizer (SHO), and Emperor Penguin Optimizer (EPO). The results and data obtained from applying SGO and other mentioned algorithms on unimodal test functions, multimodal test functions, and pressure vessel design problem show that SGO is able to provide better results in comparison with other well-known optimization algorithms. Moreover, results of Wilcoxon signed rank test confirm that SGO achieves more accuracy in comparison with the mentioned algorithms.

Keywords: Shell, Shell game, Shell game optimization, Optimization, Game-based algorithms.

1. Introduction

In recent years, various algorithms have been presented in the literature in order to solve optimization problems [1-6]. Optimization algorithms are applied by researchers in various fields of science and technology such as energy [5, 7], power engineering [8-10], energy carriers [11, 12], and protection [13]. Population-based algorithms can be generally classified into four categories including Physics-based, Evolutionary-based, Swarm-based, and Game-based algorithms.

1.1 Physics-based algorithms

These algorithms have been developed using the rules of physics. Simulated Annealing (SA) is based on the gradual freezing technique. The gradual freezing technique is a way to achieve a state, in which solid-state energy is minimized well and uniformly. This technique involves placing the substance at high temperature and then gradually lowering it [14]. Spring Search Algorithm (SSA) is inspired by Hooke's law. In SSA, search agents are a group of weights, which are connected together with springs [3]. Some of the other popular physics-based algorithms are Gravitation Search Algorithm (GSA) [15], Charged System Search (CSS) [16], Galaxy-based Search Algorithm (GbSA) [17], Curved Space Optimization (CSO) [18], Ray Optimization (RO) algorithm [19], Artificial Chemical Reaction Optimization Algorithm (ACROA) [20], Small World Optimization Algorithm (SWOA) [21],

Central Force Optimization (CFO) [22], Black Hole (BH) [23], and Big-Bang Big-Crunch (BBBC) [24].

1.2 Evolutionary-based algorithms

These algorithms combine aspects of natural selection and continuity of coordination. An evolutionary algorithm protects the population from the structures of the selection rules, recombination, change, and survival. These structures are based on genetic operators. In this method, the environment determines the coordination or performance of each population, and uses more consistent individuals to reproduce. Genetic Algorithm (GA) is one of the most popular evolutionary-based algorithms. GA simulates the genetic evolution of living organisms [25]. Another evolutionary-based algorithm is Differential Evolution (DE) that was presented to overcome the main flaw of the GA, the lack of local search. The main difference between GA and DE is in the selection operator [26]. Some of the other Evolutionary-based algorithms are Evolution Strategy (ES) [27], Genetic Programming (GP) [28], and Biogeography-based Optimizer (BBO) [29].

1.3 Swarm-based algorithms

These techniques are inspired by the natural processes of plants, foraging behaviors of insects, and social behaviors of animals [30]. Particle Swarm Optimization (PSO) is in this category that simulates the bird's behavior [31]. Ant Colony Optimization (ACO) is inspired by the ability of the ants to find the shortest route between the nest and a food source [32]. Some of the other Swarm -based algorithms are Artificial Bee Colony (ABC) [33], Bat-inspired Algorithm (BA) [34], Spotted Hyena Optimizer (SHO) [35], Cuckoo Search (CS) [36], Emperor Penguin Optimizer (EPO) [37], Grey Wolf Optimizer (GWO) [38], Grasshopper Optimization Algorithm (GOA) [39], Group Optimization (GO) [40], 'Following' Optimization Algorithm (FOA) [41], and Donkey Theorem Optimization (DTO) [42].

1.4 Game-based algorithms

These algorithms are developed based on the rules of various games. Dehghani et al. suggested Orientation Search Algorithm (OSA), which is inspired by the rules of the orientation game. In this game, players move in the orientation of the referee's hand [1, 43]. Dice Game Optimizer (DGO) is another game based algorithm that simulate an old game entitled dice game [44].

1.5 Contribution

So far, many algorithms have been proposed by researchers in the first three categories (Physics-based, Evolutionary-based, and Swarm-based algorithms), which are applied in various fields of science. The main idea of these algorithms is using the nature of different phenomena to achieve a common goal. Since players strive to achieve a goal (called victory) in various individual and group games, the rules of these games are also very useful to design optimization algorithms. In this regard, the contribution of the authors is proposing a new game-based optimization technique.

This paper presents a novel game-based algorithm entitled Shell Game Optimization (SGO) for solving the optimization problems. SGO is inspired by the rules governing on a game called shell game. Shell game is based on the precision and intelligence that each player should find the shell, under which the object is hidden. Many of the mentioned optimization algorithms encounter with two challenges, setting of multiple control parameters and complexity of the equations. However, lack of control parameters and simplicity of the equations as well as implementation are the important features of SGO. Therefore, SGO can be easily applied to any optimization problem. The performance of SGO has been compared to eight well-known optimization techniques considering twenty-three linear and nonlinear benchmark test functions. Moreover, SGO has been tested on an engineering optimization problem to validate its effectiveness.

1.6 Paper structure

The rest of this paper is organized as follows: Section 2 describes the shell game. SGO is explained in section 3. The experimental results and discussion are presented in section 4. Finally, the conclusion is given in section 5.

2. Shell game

Shell game is an old game, in which the operator provides three shells and a small ball as shown in Fig. 1. In this game, the curiosity of players is stimulated, which helps to increase the accuracy of the players. First, the operator invites several persons as players. Then the operator shows the ball to the players. After that, puts the ball under one of the shells. The operator moves the shells on the table using hand

Figure. 1 Shell game

gestures. Now the operator asks the players to guess the shell under which the ball is hidden. Each player may choose the correct or wrong shell, depending on the degree of accuracy and intelligence. More points are awarded to the player that recognizes the correct shell.

In this paper, a new optimization method is introduced inspired by this game.

3. Shell Game Optimization (SGO)

In this section, shell game is simulated to invent a new optimization algorithm called Shell Game Optimization (SGO). For this purpose, the following assumptions are considered:

- In this game, a person is considered as the game's operator.
- Three shells and one ball are available to the operator.
- Each player has only two opportunities to guess the correct shell.

3.1. Mathematical Model

Now, a set of *N* person is assumed as the game's players. In Eq. (1), the position '*d*' of player '*i*' is shown as x_i^d.

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

Here, X_i is actually a random value for the problem variables. Based on X_i, the value of the fitness function is evaluated for each player.

After calculating the fitness function value for each player, the game's operator chooses three shells that one of the shells is related to the position of the best player and two other shells is chosen randomly by Eq. (2).

$$game's\ operator: \begin{cases} shell_1 = ball = X_{best} \\ shell_2 = X_{k_1} \\ shell_3 = X_{k_2} \end{cases} \tag{2}$$

Where, X_{best} is the position of minimum (in minimization problems) or maximum (in maximization problems) of fitness, X_{k_1} and X_{k_2} are positions of two members of the population. k_1 and k_2 are random numbers between 1 to *N*, which are chosen randomly.

After calculating the fitness function and identifying the shells for each player, intelligence and accuracy of the players should be evaluated in this stage. Each player guesses the shell based on accuracy and intelligence. Accuracy and intelligence of each player are simulated according to the fitness normalized value by Eq. (3).

$$AI_i = \frac{fit_i - fit\ (X_{worst})}{\sum_{j=1}^{N}[fit_j - fit\ (X_{worst})]} \tag{3}$$

Where AI_i is the accuracy and intelligence of player i and X_{worst} is the position of minimum (in maximization problems) or maximum (in minimization problems) of fitness.

Now, the player is ready to guess the ball. Given that the game is played with three shells and each player has only two chances, there are three states of guess for each player. In the first state, the first guess may be correct and the location of the ball will be recognized. In the second state, the player after a wrong guess in the first selection may guess the ball's location in the second time. Finally, in the third state, both guesses of player may be wrong and thus the player was unsuccessful to recognize the ball's location. The guess vector specified by G_v is simulated by Eq. (4) for each player.

$$G_v(x) = \begin{cases} state\ 1: [1\ 0\ 0], & at\ first \\ state\ 2: \begin{cases} [0.5\ 0.5\ 0] \\ [0.5\ 0\ 0.5] \end{cases}, & at\ second \\ state\ 3: [0\ 0.5\ 0.5], & else \end{cases} \tag{4}$$

The probability of choosing one of the states for shell selection is simulated by Eq. (5).

$$state = \begin{cases} state\ 1:\ if\ AI_i > r_{g_1} \\ state\ 2:\ if\ AI_i > r_{g_2} \\ state\ 3:\ else \end{cases} \tag{5}$$

Where r_{g_1} is the possibly of correct guess at the first selection and r_{g_2} denotes the possibly of correct guess at the second time.

Finally, X_i vector, which is assumed as the location of each member of population, is updated according to Eqs. (6)-(9).

$$dx_{i,ball}^{d} = r_1 \times (ball - x_i^d) \times \text{state (1,1)} \quad (6)$$

$$dx_{i,shell_2}^{d} = r_2 \times \left(shell_2^d - x_i^d\right) \times sign(fit_i - fit_{shell_2}) \times \text{state (1,2)} \quad (7)$$

$$dx_{i,shell_3}^{d} = r_3 \times \left(shell_3^d - x_i^d\right) \times sign(fit_i - fit_{shell_3}) \times \text{state (1,3)} \quad (8)$$

$$x_i^d = x_i^d + dx_{i,ball}^{d} + dx_{i,shell_2}^{d} + dx_{i,shell_3}^{d} \quad (9)$$

Where r_i is a random value in the range of [0 1], $dx_{i,ball}^{d}$, $dx_{i,shell_2}^{d}$, and $dx_{i,shell_3}^{d}$ are the displacements of dimension '*d'* of player '*i'* based on $shell_1$, $shell_2$, and $shell_3$.

3.2. Steps of SGO

The steps of SGO are summarized as follows:
Step 1: Random formation of initial population using Eq. (1)
Step 2: Calculating the fitness value of agents
Step 3: Selection of *i*-th member
Step 4: Selecting three shells using Eq. (2)
Step 5: Calculation of accuracy and intelligence (*AI*) using Eq. (3)
Step 6: Simulating the state of guess using Eqs. (4) and (5)
Step 7: Selection of *d*-th dimension of *i*-th member
Step 8: Calculating $dx_{i,ball}^{d}$, $dx_{i,shell_2}^{d}$, and $dx_{i,shell_3}^{d}$ using Eqs. (6)-(8)
Step 9: Updating location of *d*-th dimension of *i*-th member using Eq. (9)
Step 10: If all dimensions of *i*-th member are updated, going Step 11, else returning Step 7
Step 11: If all members are updated, going Step 12, else returning Step 3
Step 12: If the stop condition is established, going Step 13, else returning Step 2
Step 13: Printing the best optimal solution

4. Experimental results and discussion

This section describes the experimentation on twenty-three standard benchmark test functions to evaluate the performance of SGO. The detailed description of these benchmarks is presented in the following. Moreover, the results of SGO are compared with eight optimization algorithms.

4.1 Benchmark test functions

The standard benchmark test functions utilized in this section have been taken from [45].

4.2 Algorithms used for comparison

Performance of the SGO algorithm is compared with the following eight optimization algorithms.

- **Genetic Algorithm (GA) [46]:** GA is inspired by genetic science and Darwinian evolution based on the survival of the highest or the natural selection. A common use of GA is its utilization as an optimization function.
- **Particle Swarm Optimization (PSO) [47]:** In PSO, the movement of the bird group is simulated as part of a sociological study that studies the concept of collective intelligence in the biological community.
- **Gravitational Search Algorithm (GSA) [15]:** GSA is inspired by law of gravity in the nature. In this algorithm, search agents are a set of objects that can be thought as planets of a system.
- **Teaching Learning Based Optimization (TLBO) [48]:** TLBO is based on teaching and learning, which is divided into two phases. The first phase, which includes learning from the teacher, and the second phase, where students learn from each other's interaction.
- **Grey Wolf Optimizer (GWO) [38]:** GWO is a nature-inspired algorithm based on the hierarchical structure and wolf's social behavior during hunting.
- **Grasshopper Optimization Algorithm (GOA) [39]:** 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) [35]:** 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) [37]:** EPO simulates the behavior of the emperor's penguins.

4.3 Performance comparison

In order to demonstrate the effectiveness of SGO, it is compared with eight well-known optimization algorithms considering unimodal, multimodal, and fixed-dimension multimodal benchmark test functions [45].

The experimentation has been done on Matlab R2014a (8.3.0.532) version in the environment of Microsoft Windows 7 using 64 bit Core i-7 processor with 2.40 GHz and 16 GB main memory. The average (Ave) and standard deviation (std) of the best optimal solution are mentioned in the tables. For each benchmark test function, SGO utilizes 30 independent runs, in which each run employs 1000 iterations.

4.3.1. Evaluation of unimodal test functions

Functions F1 to F7 are Unimodal test functions. The average results obtained during 20 times independent implementation of the algorithms are presented in Table 1. The results indicate that the SGO performance is better than other algorithms for all of the mentioned functions (F1 to F7) [45].

4.3.2. Evaluation of multimodal test functions

In multimodal test functions, the number of local responses increases exponentially with increase of the function dimensions. Therefore, it is hardly possible to achieve the minimum answer for this type of functions. In this type of functions, reaching the nearest answer indicates the remarkable capability of the algorithm for passing the wrong local answers. The results of evaluating functions F8 to F13 [45] for 20 independent runtimes are presented in Table 2. For all of these functions, SGO has achieved a better performance.

4.3.3. Evaluation of multimodal test functions with low dimension

Functions F14 to F23 in [45] have a low number of dimensions and also low local answers. The results of 20 times implementation of SGO and other algorithms for these multimodal test functions are presented in Table 3. These results show that SGO also performs effectively for this type of functions and is very competitive over other optimization algorithms. Convergence curves of SGO and other optimization algorithms for three models of the functions are illustrated in Fig. 2. For unimodal functions such as F_5, multimodal test functions with high dimension such as F_{12}, and multimodal test functions with low dimension such as F_{15}, SGO converges with more precision and speed in the search space due to its adaptive mechanism.

4.3.4. Pressure vessel design

In this section, SGO has been applied on an engineering design problem. Mathematical model ofthis problem has been taken from [49]. Tables 4 and 5 show the performance of SGO and other algorithms.

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

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	SGO
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}	6.74×10^{-35}
	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}	9.17×10^{-36}
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}	7.78×10^{-45}
	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}	3.48×10^{-45}
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}	2.63×10^{-25}
	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}	9.83×10^{-27}
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}	4.65×10^{-26}
	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}	4.68×10^{-29}
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	5.41×10^{-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}	5.05×10^{-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}	8.03×10^{-24}
	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}	5.22×10^{-26}
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}	3.33×10^{-8}
	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}	1.18×10^{-6}

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

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	SGO
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	$4.37\times10^{+1}$	$4.28\times10^{+1}$	$5.72\times10^{+1}$	$2.83\times10^{+1}$	$9.19\times10\times10^{+1}$	$3.94\times10^{+1}$	$9.32\times10^{+1}$	$5.92\times10^{+1}$	9.14×10^{-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}	8.76×10^{-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}	4.85×10^{-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}	8.04×10^{-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}	3.34×10^{-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}	4.23×10^{-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}	5.11×10^{-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}	6.33×10^{-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}	4.71×10^{-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 SGO and other algorithms considering Multimodal test functions with low dimension.

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	SGO
F_{14}	Ave	4.39	2.77	3.61	6.79	$9.98\times10^{+1}$	1.26	3.71	1.08	9.98×10^{-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}	7.64×10^{-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}	3.3×10^{-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}	1.25×10^{-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}	5.12×10^{-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}	4.56×10^{-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}	1.15×10^{-18}
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}	5.61×10^{-10}
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}	4.29×10^{-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	1.25×10^{-2}
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}	3.65×10^{-7}
F_{23}	Ave	-3.10	-9.19	-9.37	-2.46	-8.41	-4.49	$-1.05\times10^{+1}$	-3.41	-10.53
	std	2.37	2.52	2.75	1.19	3.13	1.96	1.81×10^{-4}	1.11×10^{-2}	5.26×10^{-6}

DOI: 10.22266/ijies2020.0630.23

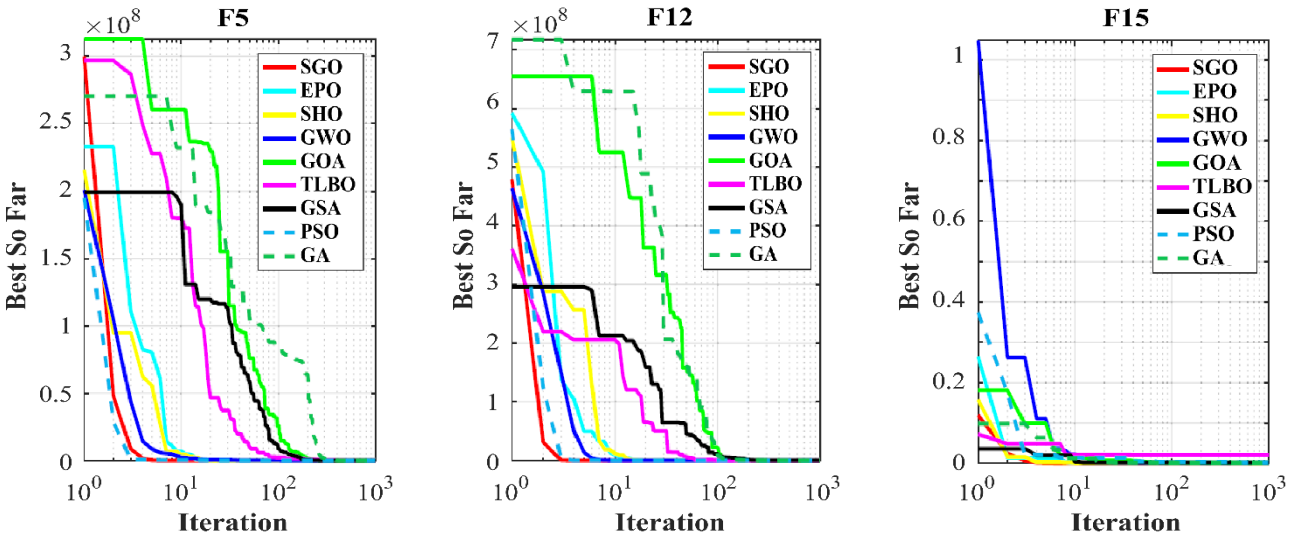

Figure. 2 Convergence curves of SGO and other optimization algorithms on three benchmark test functions

Table 4. Comparison of results for pressure vessel design problem

Algorithms	Optimum variables				Optimum cost
	T_s	T_h	R	L	
SGO	0.778099	0.383241	40.315121	200.00000	**5880.0700**
EPO	0.778210	0.384889	40.315040	200.00000	5885.5773
SHO	0.779035	0.384660	40.327793	199.65029	5889.3689
GWO	0.778961	0.384683	40.320913	200.00000	5891.3879
GOA	0.845719	0.418564	43.816270	156.38164	6011.5148
TLBO	0.817577	0.417932	41.74939	183.57270	6137.3724
GSA	1.085800	0.949614	49.345231	169.48741	11550.2976
PSO	0.752362	0.399540	40.452514	198.00268	5890.3279
GA	1.099523	0.906579	44.456397	179.65887	6550.0230

Table 5. Statistical results for pressure vessel design problem

Algorithms	Best	Mean	Worst	Std. Dev.	Median
SGO	5880.0700	5884.1401	5891.3099	024.341	5883.5153
EPO	5885.5773	5887.4441	5892.3207	002.893	5886.2282
SHO	5889.3689	5891.5247	5894.6238	013.910	5890.6497
GWO	5891.3879	6531.5032	7394.5879	534.119	6416.1138
GOA	6011.5148	6477.3050	7250.9170	327.007	6397.4805
TLBO	6137.3724	6326.7606	6512.3541	126.609	6318.3179
GSA	11550.2976	23342.2909	33226.2526	5790.625	24010.0415
PSO	5890.3279	6264.0053	7005.7500	496.128	6112.6899
GA	6550.0230	6643.9870	8005.4397	657.523	7586.0085

Table 6. Wilcoxon signed rank tests for unimodal functions F_1-F_7

	EPO	**SHO**	**GWO**	**GOA**	**TLBO**	**GSA**	**PSO**	**GA**
F_1	-1	-1	-1	-1	-1	-1	-1	-1
F_2	-1	-1	-1	-1	-1	-1	-1	-1
F_3	-1	-1	-1	-1	-1	-1	-1	-1
F_4	-1	-1	-1	-1	-1	-1	-1	-1
F_5	-1	-1	-1	-1	-1	-1	-1	-1
F_6	-1	-1	-1	-1	-1	-1	-1	-1
F_7	-1	-1	-1	-1	-1	-1	-1	-1

SGO provides optimal solution at (0.778099, 0.383241, 40.315121, 200.00000) with corresponding fitness value equal to 5880.0700.

4.3.5. Wilcoxon Signed Rank Test

Wilcoxon signed rank test [50] is used to compare the data in two groups dependent on each other.

Table 7. Wilcoxon signed rank tests for multimodal functions F_8-F_{13}

	EPO	SHO	GWO	GOA	TLBO	GSA	PSO	GA
F_8	-1	-1	-1	-1	-1	-1	-1	-1
F_9	-1	-1	-1	-1	-1	-1	-1	-1
F_{10}	-1	-1	-1	-1	-1	-1	-1	-1
F_{11}	-1	-1	-1	-1	-1	-1	-1	-1
F_{12}	-1	-1	-1	-1	-1	-1	-1	-1
F_{13}	-1	-1	-1	-1	-1	-1	-1	-1

Table 8. Wilcoxon signed rank tests for multimodal functions F_{14}-F_{23}

	EPO	SHO	GWO	GOA	TLBO	GSA	PSO	GA
F_{14}	-1	-1	-1	-1	-1	-1	-1	-1
F_{15}	-1	-1	-1	-1	-1	-1	-1	-1
F_{16}	-1	-1	-1	-1	-1	-1	-1	-1
F_{17}	-1	-1	-1	-1	-1	-1	-1	-1
F_{18}	-1	-1	-1	-1	-1	-1	-1	-1
F_{19}	-1	-1	-1	-1	-1	-1	-1	-1
F_{20}	-1	-1	-1	-1	-1	-1	-1	-1
F_{21}	-1	-1	-1	-1	-1	-1	-1	-1
F_{22}	-1	-1	-1	-1	-1	-1	-1	-1
F_{23}	-1	-1	-1	-1	-1	-1	-1	-1

Table 9. Wilcoxon signed rank tests for pressure vessel design (PVD) problem.

	EPO	SHO	GWO	GOA	TLBO	GSA	PSO	GA
PVD	-1	-1	-1	-1	-1	-1	-1	-1

Based on the fitness function, the Wilcoxon test was performed at 95% confidence level (the zero hypothesis in this test indicates lack of difference and the opposite hypothesis indicates the difference), and the results show that SGO achieves more accuracy in comparison with the mentioned eight algorithms. Wilcoxon signed rank test results are presented in Tables 6 to 9. In these tables, -1 means worse, 0 means equal, and 1 means better.

5. Conclusion

In this paper, a novel optimization method entitled Shell Game Optimization was introduced. SGO is based on the rules of the Shell game. In this game, players try to find a ball that is hidden under one of the three Shells. SGO and eight other optimisation algorithms were tested on 23 benchmark test functions. In addition, pressure vessel design problem was considered to further evaluate the effectiveness of the proposed algorithm. The results demonstrate that SGO has good performance compared to GA, PSO, GSA, TLBO, GWO, GOA, SHO, and EPO. Nevertheless, SGO was also analyzed considering the Wilcoxon signed rank test. Based on the results obtained for SGO and other listed optimization algorithms; it was shown that SGO is able to handle different types of constraints very efficiently and provides better solutions. The results obtained for unimodal and multimodal test functions confirmed the superior exploitation and exploration capability of SGO.

For future works, there are several ideas that is suggested by the authors for study. As an interesting future contribution, one can develop a binary version of SGO. In addition, SGO can be applied to solve many-objective real-life optimization as well as multi-objective problems.

Acknowledgments

J. M. Guerrero was funded by a Villum Investigator grant (no. 25920) from The Villum Fonden.

References

[1] 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.

[2] 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)*, pp. 220-224, 2017.

[3] 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)*, pp. 210-214, 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, 2020.

[5] 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.

[6] 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.

[7] 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.

[8] 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)*, pp. 215-219, 2017.

[9] 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.

[10] A. Suvorov, A. Gusev, N. Ruban, M. Andreev, A. Askarov, and S. Stavitsky, "The hybrid real-time dispatcher training simulator: basic approach, software-hardware structure and case study", *International Journal of Emerging Electric Power Systems*, Vol. 20, 2019.

[11] 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.

[12] 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)*, pp. 539-543, 2017.

[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)*, pp. 1372-1377, 2017.

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

[15] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences*, Vol. 179, pp. 2232-2248, 2009.

[16] A. Kaveh and S. Talatahari, "A Novel Heuristic Optimization Method: Charged System Search", *Acta Mechanica*, Vol. 213, pp. 267-289, 2010.

[17] 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.

[18] 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.

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

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

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

[22] R. A. Formato, "Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization", In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, ed: Springer, pp. 221-238, 2008.

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

[24] O. K. Erol and I. Eksin, "A New Optimization Method: Big Bang–Big Crunch", *Advances in Engineering Software*, Vol. 37, pp. 106-111, 2006.
[25] 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.
[26] 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.
[27] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies–A Comprehensive Introduction", *Natural Computing*, Vol. 1, pp. 3-52, 2002.
[28] G. M. Khan, "Evolutionary Computation", In: *Evolution of Artificial Neural Development*, ed: Springer, pp. 29-37, 2018.
[29] S. Mirjalili, "Biogeography-Based Optimisation", In: *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 57-72, 2019.
[30] S. M. Lim and K. Y. Leong, "A Brief Survey on Intelligent Swarm-Based Algorithms for Solving Optimization Problems", In: *Nature-inspired Methods for Stochastic, Robust and Dynamic Optimization*, ed: IntechOpen, 2018.
[31] J. C. Bansal, "Particle Swarm Optimization", In: *Evolutionary and Swarm Intelligence Algorithms*, ed: Springer, pp. 11-23, 2019.
[32] S. Mirjalili, "Ant Colony Optimisation", In: *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 33-42, 2019.
[33] D. Karaboga and B. Basturk, "On the Performance of Artificial Bee Colony (ABC) Algorithm", *Applied Soft Computing*, Vol. 8, pp. 687-697, 2008.
[34] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, ed: Springer, pp. 65-74, 2010.
[35] 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.
[36] 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.
[37] G. Dhiman and V. Kumar, "Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems", *Knowledge-Based Systems*, Vol. 159, pp. 20-50, 2018.
[38] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.
[39] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and Application", *Advances in Engineering Software*, Vol. 105, pp. 30-47, 2017.
[40] M. Dehghani, Z. Montazeri, A. Dehghani, and O. P. Malik, "GO: Group Optimization", *Gazi University Journal of Science*, Vol. 33, 2020.
[41] 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.
[42] M. Dehghani, M. Mardaneh, O. P. Malik, and S. M. NouraeiPour, "DTO: Donkey Theorem Optimization", In: *Proc. of Iranian Conference on Electrical Engineering (ICEE)*, pp. 1855-1859, 2019.
[43] 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.
[44] M. Dehghani, Z. Montazeri, O. P. Malik, "DGO: Dice Game Optimizer", *Gazi University Journal of Science*, Vol. 32, pp. 871-882, 2019.
[45] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82-102, 1999.
[46] S. Mirjalili, "Genetic Algorithm", In: *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp.43-55, 2019.
[47] S. Mirjalili, "Particle Swarm Optimisation", In: *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp.15-31, 2019.
[48] 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.
[49] B. Kannan and S. N. Kramer, "An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design", *Journal of Mechanical Design*, Vol. 116, pp. 405-411, 1994.
[50] R. Woolson, "Wilcoxon Signed-Rank Test", *Wiley Encyclopedia of Clinical Trials*, pp. 1-3, 2007.