Expert Systems With Applications 134 (2019) 178–191

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Social mimic optimization algorithm and engineering applications

Saeed Balochian[a,*], Hossein Baloochian[b]

[a] Department of Electrical Engineering, Gonabad Branch, Islamic Azad University, Gonabad, Khorasan-e-Razavi 9691664791, Iran
[b] Road Maintenance & Transportation Organization, Iran

ARTICLE INFO

Article history:
Received 12 August 2018
Revised 10 May 2019
Accepted 25 May 2019
Available online 27 May 2019

Keywords:
Meta-heuristic
Optimization
Swarm intelligence
Social mimic optimization

ABSTRACT

Increase in complexity of real world problems has provided an area to explore efficient methods to solve computer science problems. Meta-heuristic methods based on evolutionary computations and swarm intelligence are instances of techniques inspired by nature. This paper presents a novel social mimic optimization (SMO) algorithm inspired by mimicking behavior to solve optimization problems. The proposed algorithm is evaluated using 23 test functions. Obtained results are compared with 14 known optimization algorithms including Whale optimization algorithm (WOA), Grasshopper optimization algorithm (GOA), Particle Swarm Optimization (PSO), Stochastic fractal search (SFS), Grey Wolf Optimizer (GWO), Optics Inspired Optimization (OIO), League Championship Algorithm (LCA), Wind Driven Optimization (WDO), Harmony search (HS), Firefly Algorithm (FA), Artificial Bee Colony (ABC), Biogeography Based Optimization (BBO), Bat Algorithm (BA), and Teaching Learning Based Optimization (TLBO). Obtained results indicate higher capability of the SMO algorithm in solving high-dimensional decision variables. Furthermore, SMO is used to solve two classic engineering design problems. Three important features of SMO are simple implementation, solving optimization problems with minimum population size and not requiring control parameters. Results of various evaluations show superiority of the proposed method in finding the optimal solution with minimum function evaluations. This superiority is achieved based on reducing number of initial population. The proposed method can be applied to applications like automatic evolution of robotics, automatic control of machines and innovation of machines in finding better solutions with less cost.

1. Introduction

Optimization is a procedure used to obtain optimal parameters of a problem with all possible values to increase or decrease the output. Complicated problems like scheduling, data clustering, image processing and adjusting neural networks are examples of optimization problems. In order to solve such problems, several classic and meta-heuristic optimization approaches have been proposed. There are lots of controversy about incapability of classic methods to obtain an optimal solution in an acceptable time. On the other hand, meta-heuristic methods have become increasingly popular in solving optimization problems due to their simplicity, flexibility, no-derivative mechanism and avoiding local optima (Mirjalili, Mirjalili, & Lewis, 2014).

Meta-heuristic methods can be divided into two single-solution and multiple-solution approaches. In the first category, search procedure is started with one solution. This unique candidate solution is enhanced in the subsequent iterations. On the contrary, multiple-solution meta-heuristic methods perform optimization using a set of random initial solutions. Population is enhanced in the subsequent iterations. According to the studies, methods based on multiple solutions are more popular than single-solution methods (Mirjalili & Lewis, 2013).

Multiple-solution methods avoid local optima due to high number of initial random solutions. These methods investigate a wider span of the search space, hence probability of finding a desired optimal solution is higher. Information of the search space can be exchanged among multiple solutions as a result of which it converges towards the optimum faster. Multiple-solution based methods require investigating more fitness functions compared to single-solution based methods. Single-solution based methods like hill-climbing (Davis, 1991) and simulated annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) follow a similar idea. But simulated annealing outperforms hill-climbing in terms of avoiding local optima. A class of multiple-solution meta-heuristic methods are methods inspired by social behavior and swarm intelligence of human, animals, herds, teams or any class of creatures. Movement of organisms as a group like bird flocks, group hunting or fish school are the motivation of particle swarm. Group is represented with particle swarm and particle swarm employs position of particles in

* Corresponding author.
E-mail address: saeed.balochian@gmail.com (S. Balochian).

https://doi.org/10.1016/j.eswa.2019.05.035

the search space to represent possible solutions of the optimization problem. Particle swarm optimization performs optimization by manipulating movement of these particles. PSO (Marini & Walczak, 2015) which is known to different scientists as an efficient optimization method is one of such methods. Another algorithm of this category which was introduced by Dorigo is Ant Colony Optimization (ACO) (Dorigo, 1992). This algorithm is inspired by social behavior of ants in a colony. Imitating swarm intelligence of ants in finding shortest path between nest and food source is the basis of ACO. In this meta-heuristic method, possible solutions of optimization problem are represented by paths between colony and food resource. Ants communicate with each other through pheromone and inform others about location of food resource. When an ant finds a food, it marks the path using pheromone based on quality of the food resource. As a path is selected by other ants for a higher number of times, pheromone density increases. Increase in density of pheromone indicates that the path is shorter and quality of the food resource is higher.

One of the most widely studied organisms in swarm intelligence is honey bee (Parpinelli & Lopes, 2011). Marriage is presented in honey bee optimization (Abbass, 2001). In this algorithm, mating flight of the queen bee is represented as transit in the search space. Mating probability is determined by speed and energy of the queen and optimality of the male honey bee. Artificial Bee Colony Optimization was first proposed by Karaboga and Basturk (2007). ABC classifies bees of the hive into three classes including scouts which fly randomly without leader, employed bees which search position of their neighbors and onlooker bees which employ population optimality to select the solution. The algorithm establishes balance between exploration and exploitation using scout bees and employed bees for local search and onlooker bees for global search. Shortcoming of exploitation equations compared to explorations in ABC algorithm is the idea of the presenting Modified Gbest Artificial Bee Colony Algorithm (Bhambu, Sharma, & Kumar, 2018).

Krishnanand and Ghose presented glowworm optimization (Krishnanand & Ghose, 2005) based on behavior of firefly. Yang (2010a) presented a novel algorithm based on firefly technique. Another algorithm is developed based on firefly behavior which is called modified firefly algorithm (Tilahun & Ong, 2012). In this algorithm, random movement of brighter firefly is modified through generating random directions instead of determining best direction in which light increases. Algorithms (Cuevas, Cienfuegos, Zaldívar, & Pérez-Cisneros, 2013; Yu & Li, 2015) for optimization are inspired by mating behavior and feeding strategy of social spider. Team coaching in volleyball league games is the basis of VPL algorithm (Moghdani & Salimifard, 2018).

Soccer is a team game in which players try to find the best position and attain a goal under the supervision of a coach. Soccer game metaheuristic algorithm simulates behavior of players (Fadakar & Ebrahimi, 2016). Importance of this method is in modeling human's intelligence. Gong employs adaptive learning based on contradiction to improve fireworks algorithm (Gong, 2016). Wang et.al. have proposed a metaheuristic algorithm called monarch butterfly optimization algorithm based on migration of the monarch butterfly (Wang, Deb, & Cui, 2015). A metaheuristic algorithm which is an optimization sine-cosine algorithm based on population is proposed in Mirjalili (2016). In this algorithm, multiple initial random candidate solutions are created. Then a mathematical model based on sine and cosine function is used to move candidate solution towards optimal solution. Cheng et.al. have proposed an algorithm to solve engineering problems inspired by symbiosis organisms (SOS). In this paper, interplay strategies of adaptive symbiosis is modeled using organisms for survival (Cheng & Prayogo, 2014). The idea proposed in whale optimization algorithm is inspired by strategy of humpback whale hunting (Mirjalili & Lewisa, 2016). Ability of Bats to manipulate frequency and wavelength of audio waves emitted while hunting is the idea of proposing dynamic virtual bats (Topal & Altun, 2016). In this algorithm, search is based on developed role to enhance bat algorithm. Combining Harmonic search algorithm with variable population and quantum principles has been proposed in Meryem and Abdelmadjid (2016). Authors have employed the proposed method to solve vehicles routing problem. Structural optimization of frequency limitation includes searching a convex and nonlinear space with multiple local optima. In various structures, natural frequency of the structure should be kept far from the excitation frequency. As these two frequencies get close to each other, the structure is affected destructively. In Dede and Togan (2015), shape and size of the structure are optimized considering frequency constraints using TLBO algorithm. Authors of Topal, Dede, and Öztürk (2017) have proposed a TLBO-based technique for optimizing fundamental frequency of non-uniform multi-layer composite sheets. Results show that TLBO is more efficient than other methods in terms of execution time and number of function evaluations. Bekiroğlu, Dede, and Ayvaz (2009) have proposed a new genetic algorithm with adaptive method and new coding for solving structural problems. Other swarm intelligence techniques in the literature include Slap Swarm Algorithm (SSA) (Mirjalili et al., 2017), Grasshopper Optimization Algorithm (GOA) (Saremi, Mirjalili, & Lewis, 2017), the ant lion optimizer (ALO) (Mirjalili, 2015a), and Dolphin Echolocation (DE) (Kaveh & Farhoudi, 2013). In order to become familiar with evolutionary computations and their applications, it is recommended to study (Gong, Wang, Liu, Yan, & Jiao, 2016).

All evolutionary and swarm intelligence algorithms require control parameters like number of population and number of generations. In addition to general parameters, some algorithms require specific control parameters. The larger is the number of these control parameters, adjusting and selecting their optimal value would be more difficult. Each algorithm might have other negative points in addition to number of control parameters. Genetic algorithm exchanges information well using mutation and cross-over operators and it is suitable for continuous problems. On the other hand, it has some shortcomings like lack of memory, inability for local search, difficult coding of problems as chromosome, computational complexity and fast convergence. PSO has resolved lack of memory and implementation difficulty but it still suffers from fast convergence, weakness in local search and being trapped in local optima. Unlike the two previous algorithms, ACO is not suitable for continuous problems and suffers from fact convergence and inability for local search. This algorithm is suitable for solving discrete problems and can explore good solutions, fast. Simple understanding and coding are two characteristics of FA, CS, ABC and AC algorithms but they are rarely studied in the literature (Adrian, Utamima, & Wang, 2014; Tan, Hassan, Shah Majid, & Abdul Rahman, 2013).

Swarm intelligence algorithms preserve information of the search space during one iteration loop. These algorithms mainly use memory to store the best obtained solution at each step. Moreover, these algorithms have less parameters; on the other hand, since less operators are used for modelling natural behavior and random search, their implementation is easier.

Despite high number of meta-heuristic algorithms, the question that "why are optimal methods required?" arises. This question is answered by Non-Free Launch theorem (Joyce & Herrmann, 2018). This theorem proves that there exists no meta-heuristic method suitable for all optimization problems. Hence, existing methods are enhanced continuously or new meta-heuristic methods are presented. Researches in the context of meta-heuristic methods improve existing methods, combine them or present novel methods. This paper presents a new meta-heuristic method based on multiple solutions inspired by mimicking behavior. The proposed method has resolved shortcoming of this class of algorithms (evaluating fitness function) by reducing number of solutions (population). The rest of this paper is organized as follows. Section 2 defines mimicking behavior from historical and scientific viewpoints. Section 3 introduces SMO algorithm. Section 4 covers experiments and results of comparing SMO with other known methods. Finally, the paper is summarized in Section 5.

2. Social mimic from historical and scientific viewpoints

Basis of imitation or assimilating morale and behavior of others can be found in early humans. Humans employ assimilation as a kind of public symptom. Human should have learned and innovated many of the accepted social behaviors to survive and evolve. There were more respectful, stronger and more intelligent human of higher rank in the society and others should have created specific behavioral patterns to show their respect for these people. For instance, if these respectful people use a bracelet for decoration, other people consider that as a conventional behavior and they ought to wear such a bracelet.

Emotional states of the ones whom we face affect our emotions. Emotions and morale like anxiety and happiness can be transferred from one person to another. This emotional imitation occurs autonomously and unconsciously, but how does science describe this synchronization in relationships? Everything returns to a neural system in human brain. Specific neurons are responsible to detect faces and percept facial expressions; interpretation of these neurons from emotional states makes you to sharpen your eyebrows or laugh. These neurons which act as a bridge among brains are responsible for unconscious imitation of operations and react when the person acts or when sensual and dynamic actions are observed in others. In other words, we are intrinsically programmed to imitate emotions expressed by others through facial expressions and body gestures. In brief, it can be said that people have always tried to assimilate behavior, speech, clothing and other characteristics of famous people. In other words, we try to move towards a better point through Social mimic behavior.

2.1. Mirror neuron system

At the end of 20th century, while studying monkeys, Rizzolatti, an Italian neuroscientist at Parma University observed that when a monkey eats a banana or observes another monkey or human eating banana, similar nerve cells are activated in his brain. Since these nerve cells reflect activities of others in our brain like mirror, scientists have called it mirror neurons (Rizzolatti & Fabbri-Destro, 2010). Since then, many researches have been done on neural system in human and other animals. Now, it is known that neural system of human is more complicated than monkeys. Subsequent observations show that 10% of inferior frontal and inferior parietal neurons mirror features and give similar responses to observed movements and behavior. These neurons provide the possibility for us to learn, understand purpose of others, their body gestures and facial expressions by observing their behavior. Laughter or frown of another individual causes laughter and frown of another individual and this can result in sympathy among people. In fact, mirror neural systems create a neural mechanism for development of emotions in human groups. In other words, our neural system is designed to understand others, communicate and sympathize with others and these circuits cannot be turned off.

According to Ramachandran, who is one of the most famous neurologists and I quote here: "I predict that mirror neural systems perform the same thing in psychology as DNA in biology" (Lacoste-Badie & Droulers, 2014).

3. Social mimic algorithm

The proposed SMO algorithm is inspired by mimicking behavior of people in society. In such behavior, each individual tries to assimilate oneself to famous people through imitating their behavior. Accordingly, in optimization problems, each solution can move towards global optimal solution by imitating parameters of the optimal solution.

In optimization problems, imitation behavior can be modelled to search solution space randomly. Each solution determines its difference with global optimal value through comparing its optimal value with the global optimal obtained in the last iteration. Then, this difference value is applied to problem parameters to search the solution space randomly. In SMO, the term "*Follower*" is used to represent population concept. Furthermore, the terms"*Leader*"and "*numberofimitations*" show best global optimal and current number of iterations. Each *Follower*represents one solution of the problem. For instance, $Follower_i$ is the i^{th} member of the population. In the following, each step of the proposed algorithm is described.

3.1. Initialization

Like other optimization algorithms, SMO starts by initializing Followers which represent set of initial solutions. Number of Followers is determined using *Pop_Size*variable which specifies population size. Each follower is represented by a $1 \times N$vector where N is the number of decision variables of the problem. Random numbers between lower bound and upper bound of each decision variable are assigned to each element of this vector using Eq. (1).

$$Follower_i(1, j) = lb_j + Rand() * (ub_j - lb_j)\, j = 1, 2, \ldots, N \tag{1}$$

$lb_j$and$ub_j$are lower and upper bounds of jth variable. Function $Rand()$generates a random number with uniform distribution between 0 and 1. In order to store set of all solutions, *Social*matrix of dimension$Pop_Size \times M$is used. Number of rows of this matrix is equal to population size and number of its columns is equal to number of decision variables plus$N+1$where$N$is the column to store values of decision variables and last column is used to store optimal value of each follower. In this step, each solution (Follower) is evaluated using fitness function and its optimality value is stored in the last column of the social matrix. Finally, the Follower with best optimality value is selected as the Leader.

3.2. Mimicking

In this step of the algorithm, each follower calculates its difference from leader obtained in the previous step using Eq. (2).

$$Difference = (Leader - FitnessOf(Follower_i)) / FitnessOf(Follower_i) \tag{2}$$

If difference value is zero, $Rand()$function is used to assign a random number in (0, 1], to it. In the next step, new values of decision variables of $Follower_i$ are calculated using Eq. (3).

$$Follower_i = Follower_i + Difference \times Follower_i \tag{3}$$

Now, optimality of each follower is evaluated using fitness function and if its value is improved, it replaces the previous value in the Social matrix. After calculating value of all Followers, best optimality is selected and value of Leader is updated.

3.3. Termination condition

Mimicking phase is continued until termination condition is met. Termination condition can be determined using maximum

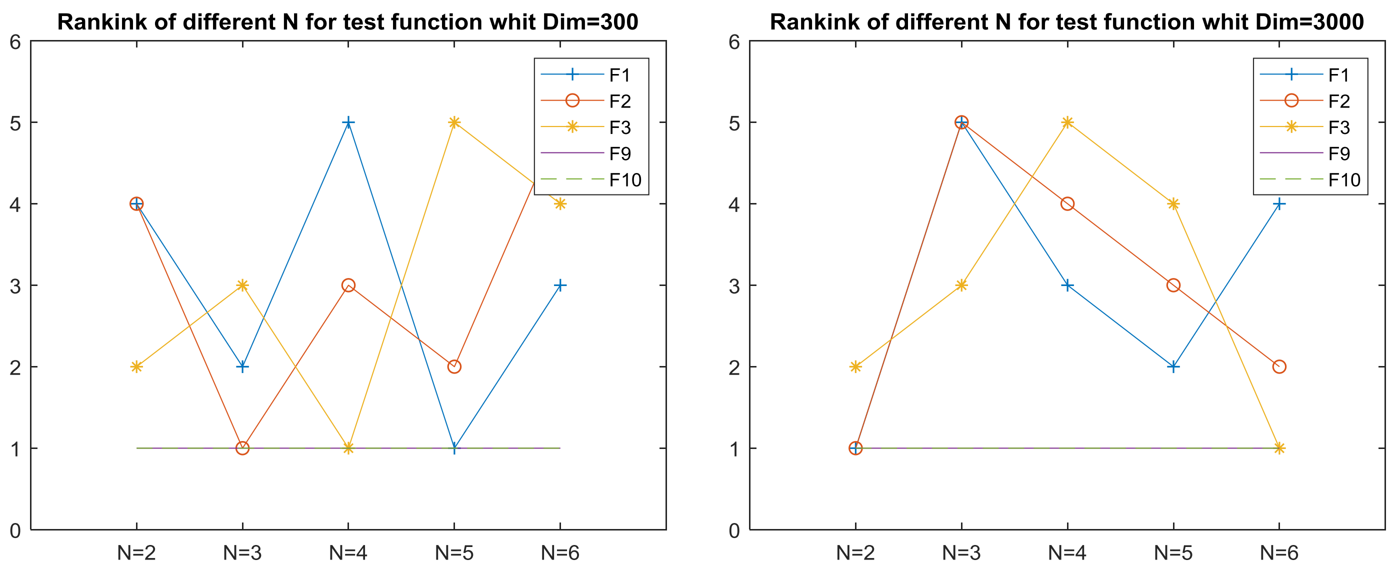

Fig 1. Ranking of N different functions.

number of iterations, maximum time of using CPU, reaching a specific error rate, maximum number of iterations without changing best optimality or any proper condition. After this step, algorithm represents the solution with best optimality. The three steps above comprise the SMO algorithm. Its pseudo-code can be seen in Algorithm 1. As can be seen in this pseudo-code, SMO does not require any initial parameters as a result of which the algorithm does not need to adjust parameter.

Algorithm 1 Social mimic algorithm.

```
Initialize the Follower population Follower_i (i = 1, 2, ..., pop_size)
Initialize Social
Calculate the fitness of each Follower
Leader = the best fitness
While (end condition not reached)
    For each Follower in Social
        Difference = (Lider − FitnessOf(Follower_i))/FitnessOf(Follower_i)
        If (Difference==0)
            Difference=-Rand(1)
        End
        For J=1:N
            Follower_i(j) = Follower_i(j) + Difference × Follower_i(j)
        End
        Check if follower goes beyond the search space and amend it
        Update Social if new fitness is better
    End
    Calculate the fitness of each follower
    Update Leader if there is a better solution
End
Return Leader
```

3.4. Characteristics of SMO

As can be seen in the pseudo-code and the above discussion, SMO does not require any initial parameters except population size. The proposed method does not require adjusting initial parameters due to the above feature. As number of initial parameters increases, proper adjustment of the parameters become an optimization problem. According to Eqs. (2) and (3), the function selected for exploration and exploitation in SMO is very simple with minimum computational complexity.

Table 1
Unimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30,300,3000	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30,300,3000	[−10,10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30,300,3000	[−100,100]	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30	[−1.28,1.28]	0

4. Results

Numerical efficiency of the SMO algorithm presented in this paper is analyzed using 23 classic benchmarks given in the optimization literature (Javidy, Hatamlou, & Mirjalili, 2015; Mirjalili, 2015b; Wang, Deb, & Coelho, 2015; Zhao & Tang, 2008). Tables 1–3 demonstrate summary of information of each function including cost function, number of decision variables, changes of optimization variables and optimal point. In general, test functions are organized in three classes including unimodal, multimodal and fixed-dimension multimodal benchmark functions. Unimodal functions ($f_1 - f_7$)are described in Table 1. These functions are suitable for evaluating efficiency of the algorithm.

Second class includes multimodal functions($f_8 - f_{13}$)which have a large number of local optimums. These functions are useful to test exploration capability of the algorithm and avoid local optima. Details of these functions are given in Table 2. Different characteristics of the third class including fixed-dimension multimodal functions are given in Table 3 which include cost function, range of variables and optimal value.

Next section investigates number of parameters of SMO algorithm and its effect. Then, results of comparing the proposed algorithm with known algorithms including Whale optimization algorithm (WOA), Grasshopper optimization algorithm (GOA), Particle Swarm Optimization (PSO), Stochastic fractal search (SFS) (Salimi, 2015), Grey Wolf Optimizer (GWO), Optics Inspired Optimization (OIO) (Husseinzadeh Kashan, 2015), League Champi-

Table 2
Multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\sqrt{\lvert x_i \rvert}$	30	[−100,100]	0
$f_9(x) = \sum_{i=1}^{n} [x_i{}^2 - 10\cos(2\pi x_i) + 10]$	30,300,3000	[−10,10]	0
$f_{10}(x) = 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30,300,3000	[−100,100]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−100,100]	0
$f_{12}(x) = \frac{\pi}{n}\left\{\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + \sin(\pi y_{i+1})] + (y_n - 1)^2\right\} +$ $\sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30	[−30,30]	0
$f_{13}(x) = 0.1\left\{\sin^2(3\pi y_1) + \sum_{i=1}^{n} \begin{matrix}(x_i - 1)^2[1 + \sin^2(3\pi y_i + 1)] + \\ (x_n - 1)^2[1 + \sin^2(2\pi x_n + 1)]\end{matrix}\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−100,100]	0

Table 3
Fixed-dimension benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_{14}(x) = \left(\frac{1}{500}\sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65,65]	1
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5,5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})(\cos(x_1)) + 10$	2	[−5,5]	0.398
$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right]$ $30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)$	2	[−2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	−3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0,10]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	4	[0,10]	−10.5363

onship Algorithm (LCA) (Husseinzadeh Kashan, 2014), Wind Driven Optimization (WDO) (Bayraktar, Komurcu, & Werner, 2010), Harmony search (HS) (Gao, Govindasamy, Xu, Wang, & Zenger, 2015), Firefly Algorithm (FA) (Fister, Fister, Yang, & Brest, 2013), Artificial Bee Colony (ABC), Biogeography Based Optimization (BBO) (Simon, 2008), Bat Algorithm (BA) (Yang, 2010b), Teaching Learning Based Optimization (TLBO) (Rao, Savsani, & Vakharia, 2011) are presented.

4.1. Effect of number of parameters on the optimization algorithm

As number of control parameters in optimization algorithms increases, adjusting and finding their proper values turns into an optimization problem. The only parameter of SMO is the population size which is common among all meta-heuristic optimization methods. SMO strategy eliminates the need to a large population size for performing exploration and exploitation. Control parameters of the algorithms selected for comparing efficiency of optimization problems are given in Table 4.

In order to evaluate effect of population size (N) on SMO algorithm, functions $f_1 f_2 f_3 f_9$ and f_{10} are selected. In order to solve the test functions, number of iterations is selected 100 and population size is considered to be 2, 3, 4, 5, 6, respectively. Each test function is solved 500 times to generate statistical results. Table 5 illustrates results obtained for test functions with 300 and 3000 decision variables.

It is obvious that$N = 2$ has generated more efficient results in most problems. According to Table 5, $N = 3$has the best ranking for functions with 300 decision variables. $N = 2$ is ranked the second. In solving problems with 3000 decision variables,$N = 2$ has the best rank for f_1, f_2, f_9 and f_{10}. Fig. 1 shows different values of population size in evaluating test functions.

According to the above results, it can be concluded that using population size of 2 or 3, the proposed algorithm is able to solve optimization problems with high number of decision variables. According to convergence curve of the response in Fig. 2, SMO has converged to a proper response in less than 10 iterations.

4.2. Comparison with other algorithms

As mentioned above, three different sets of problems are used to demonstrate efficiency of the proposed Algorithm. 14 recent and known algorithms are used for evaluation. In order to guarantee best efficiency, control parameters of other algorithms are

Table 4
Control Parameters of algorithms.

Algorithm	Parameter name
WOA	Shape of logarithmic spiral, and 3 random generated parameter
GOA	cmin, and cmax
PSO	Cognitive and social constant, Inertial weight
SFS	Gaussian walk, Maximum Diffusion Number
GWO	2 random generated parameter
OIO	Number of Light Points, random perturbation required
LCA	Lower and upper limit for retreat coefficient, Lower and limit for approach coefficient, Probability of success, alpha, and type of formation
WDO	RT coefficient, Gravitational constant, Constants in the update eq, Coriolis effect, and maximum allowed speed
HS	Harmony Memory Consideration Rate, Pitch Adjustment Rate, Fret Width (Bandwidth), Fret Width Damp Ratio
FA	Alpha, beta, and gamma
ABC	Number of food sources, value of limit, and the maximum cycle number (MCN).
BBO	Mutation probability, Number of the best solutions to keep from one generation to the next
BA	Loudness, pulse rate, frequency min and max
TLBO	Elitism in (R. Venkata Rao & Patel, 2013)
SMO	–

Table 5
Results obtained from different number of follower.

	Dim = 300	Dim = 3000								
Follower	2	3	4	5	6	2	3	4	5	6
f_1	2.35E−111	5.07E−112	6.15E−108	7.56E−113	7.42E−112	2.02E−63	1.02E−53	2.02E−57	2.90E−59	4.85E−57
f_2	2.88E−56	2.65E−58	1.44E−57	8.67E−58	3.93E−52	1.98E−32	3.71E−31	1.38E−31	5.84E−32	2.26E−32
f_3	1.24E−110	1.67E−106	8.99E−121	7.31E−104	1.90E−104	6.87E−59	6.36E−58	3.22E−56	8.07E−57	5.75E−59
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16

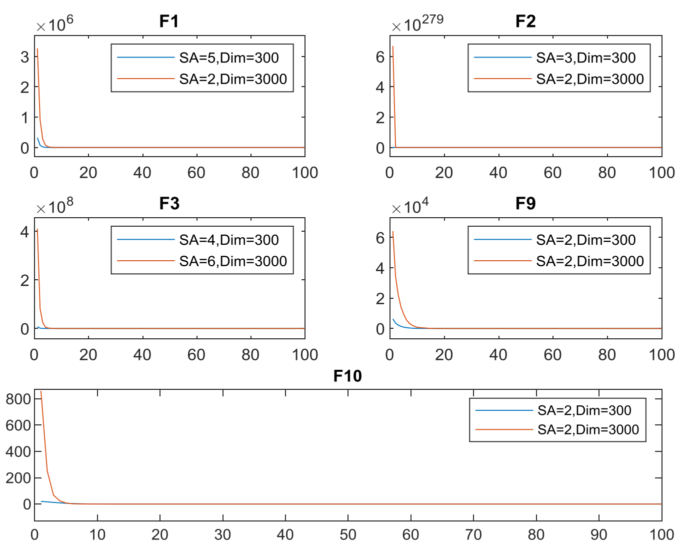

Fig 2. The convergence curve of SMO obtained in some of the benchmark functions.

selected based on their value in the last version (source code). In this test, population size and number of iterations for all algorithms are considered to be 2 and 100, respectively. Each test function is solved 100 times. In order to compare efficiency of the algorithms, in addition to the best response, worst response, mean and standard deviation of the generated responses are also presented in Tables 6–8. According to the results, SMO algorithm has generated the best response for functions $f_1 f_2 f_3 f_4 f_9 f_{10} f_{11} f_{14} f_{16}$. The worst ranked obtained by SMO is 13 for f_{18}.

Information of Tables 6–8 are summarized in Table 9. First number in each column is the number of problems for which SMO has obtained higher accuracy results compared to other algorithms. According to the represented information, results obtained using are comparable with other methods. Only SFS has generated results similar to SMO for two problems. Finally, it can be said that in solving 23 optimization problems, the proposed method has obtained more accurate results for more than half of the problems with less cost compared to other algorithms.

4.3. Solving classic engineering problems

Solving structural design problems using random optimization techniques is a popular research area. In this section, two engineering design problems are solved using SMO and the obtained results are compared with the results published in other papers.

4.3.1. Cantilever beam design problem

According to Fig. 3, cantilever beam is comprised of 5 quad square sections. Length of these 5 sections are parameters of the problem. In addition, this problem has a constraint which should not be violated by the final optimal design.

This design problem is formulated as follows.

$$\text{Consider } \overrightarrow{X} = [x_1 x_2 x_3 x_4 x_5]$$

$$\text{Minimize } f(\overrightarrow{X}) = 0.6224(x_1, x_2, x_3, x_4, x_5)$$

$$\text{Subject to } g_1(\overrightarrow{X}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0$$

$$\text{Variable range } 0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100$$

In order to solve this problem, SMO algorithm with two or five search agents and maximum number of 500 iterations is used to determine optimal values. Comparison results of ALO, SSA, GOA, SOS, cuckoo search algorithm (CS) (Gandomi, Yang, & Alavi, 2013), Generalized Convex Approximation (GCA) (Chickerman & Gea, 1996), Method of Moving Asymptotes (MMA)(Chickerman & Gea, 1996) are given in Table 10.

By investigating the results presented in Table 10, high efficiency of the proposed method in estimating optimal solution observing constraints of the problem can be seen. SMO has minimum

Table 6
Results of unimodal functions.

Function		SMO	WOA	GOA	PSO	SFS
F1	Best	7.02E−116	1.13E−06	3.25E+04	1.36E+04	2.46E−91
	Worst	1.31E−60	5.12E+03	9.46E+04	6.04E+04	3.50E−60
	Ave	1.56E−62	2.75E+02	6.03E+04	3.13E+04	3.58E−62
	Std	1.33E−61	6.99E+02	1.25E+04	8.70E+03	3.50E−61
F2	Best	1.02E−56	4.25E−07	6.18E+03	5.34E+01	5.15E−51
	Worst	9.30E−32	8.43E+00	5.87E+20	2.94E+09	1.04E−28
	Ave	9.97E−34	5.38E−01	6.66E+18	2.94E+07	1.08E−30
	Std	9.31E−33	1.17E+00	5.90E+19	2.94E+08	1.04E−29
F3	Best	2.49E−111	2.71E+04	2.96E+04	1.84E+04	6.33E−95
	Worst	1.92E−55	7.02E+05	2.54E+06	1.76E+05	1.26E−49
	Ave	1.92E−57	2.13E+05	3.52E+05	5.75E+04	1.26E−51
	Std	1.92E−56	1.39E+05	3.80E+05	2.70E+04	1.26E−50
F4	Best	1.04E−55	1.30E+00	6.14E+01	4.86E+01	8.03E−50
	Worst	1.16E−32	9.75E+01	9.72E+01	9.71E+01	6.77E−30
	Ave	1.17E−34	7.11E+01	8.41E+01	7.12E+01	1.64E−31
	Std	9.31E−33	2.65E+01	8.15E+00	9.64E+00	8.12E−31
F5	Best	2.90E+01	2.88E+01	8.98E+07	1.96E+07	2.87E+01
	Worst	2.90E+01	3.02E+07	4.14E+08	2.25E+08	2.90E+01
	Ave	2.90E+01	7.28E+05	2.29E+08	7.14E+07	2.90E+01
	Std	7.11E−03	3.14E+06	6.96E+07	3.78E+07	3.60E−02
F6	Best	6.08E+00	3.31E+00	2.88E+04	1.28E+04	4.83E+00
	Worst	7.50E+00	4.59E+03	8.64E+04	5.73E+04	7.39E+00
	Ave	7.40E+00	2.41E+02	5.89E+04	3.27E+04	6.84E+00
	Std	2.24E−01	7.08E+02	1.07E+04	9.14E+03	4.19E−01
F7	Best	1.93E−05	6.64E−03	3.84E+01	4.78E+00	9.57E−06
	Worst	4.44E−02	1.86E+01	2.85E+02	1.25E+02	1.73E−02
	Ave	1.10E−02	1.36E+00	1.70E+02	4.86E+01	4.10E−03
	Std	9.71E−03	3.28E+00	4.75E+01	2.52E+01	3.77E−03
Function		GWO	OIO	LCA	WDO	HS
F1	Best	1.02E−02	3.85E−08	1.46E+04	2.55E−02	6.45E+02
	Worst	4.91E+03	2.36E−03	5.78E+04	3.41E+02	2.90E+03
	Ave	3.83E+02	4.00E−05	3.75E+04	6.05E+01	1.64E+03
	Std	7.35E+02	2.36E−04	8.04E+03	8.38E+01	4.81E+02
F2	Best	2.56E−04	7.77E−04	6.14E+01	2.21E−02	7.75E+00
	Worst	2.03E+01	5.61E−02	5.17E+12	1.04E+01	1.89E+01
	Ave	4.16E+00	9.46E−03	5.20E+10	3.66E+00	1.32E+01
	Std	4.95E+00	8.63E−03	5.17E+11	2.26E+00	2.27E+00
F3	Best	4.55E+02	4.67E−01	4.45E+04	2.91E−01	1.13E+04
	Worst	3.08E+04	3.74E+04	2.47E+05	1.71E+03	3.83E+04
	Ave	9.45E+03	1.24E+03	9.22E+04	5.50E+02	2.46E+04
	Std	6.62E+03	3.98E+03	3.57E+04	3.87E+02	6.49E+03
F4	Best	3.65E+00	1.17E−04	6.79E+01	1.53E−01	2.87E+01
	Worst	6.13E+01	1.33E−02	9.69E+01	8.73E+00	4.52E+01
	Ave	2.81E+01	2.10E−03	8.57E+01	4.31E+00	3.80E+01
	Std	1.27E+01	1.87E−03	5.48E+00	2.19E+00	3.85E+00
F5	Best	4.02E+01	2.54E−04	1.87E+07	3.00E+01	9.99E+04
	Worst	7.29E+06	1.68E+01	2.67E+08	1.62E+04	1.15E+06
	Ave	1.76E+05	5.76E−01	1.09E+08	1.84E+03	3.38E+05
	Std	7.53E+05	2.41E+00	5.00E+07	3.07E+03	1.67E+05
F6	Best	4.60E+00	1.72E−05	1.84E+04	2.54E−01	8.37E+02
	Worst	4.76E+03	2.45E−02	6.74E+04	3.13E+02	2.99E+03
	Ave	3.30E+02	1.91E−03	3.86E+04	6.36E+01	1.68E+03
	Std	6.70E+02	3.44E−03	8.04E+03	8.13E+01	4.38E+02
F7	Best	5.82E−03	3.77E−04	1.65E+01	1.05E−02	3.00E−01
	Worst	1.90E+00	2.74E−02	1.23E+02	2.09E−01	1.82E+00
	Ave	2.72E−01	7.00E−03	5.97E+01	7.45E−02	8.62E−01
	Std	3.18E−01	5.24E−03	2.67E+01	4.01E−02	2.95E−01
Function		FA	ABC	BBO	BA	TLBO
F1	Best	2.46E+04	3.05E+04	5.94E+04	2.60E+04	1.44E−09
	Worst	8.51E+04	7.89E+04	1.14E+05	1.11E+05	7.46E−01
	Ave	5.48E+04	5.48E+04	8.99E+04	7.17E+04	3.85E−02
	Std	1.23E+04	1.19E+04	1.33E+04	1.79E+04	9.86E−02
F2	Best	7.79E+01	8.11E+01	1.09E+08	5.15E+02	1.18E−06
	Worst	7.83E+11	4.51E+11	1.70E+19	3.20E+16	1.46E−01
	Ave	1.11E+10	5.55E+09	5.00E+17	1.15E+15	2.23E−02
	Std	7.93E+10	4.51E+10	2.08E+18	4.70E+15	2.50E−02
F3	Best	3.26E+04	3.15E+04	8.26E+04	6.77E+04	6.15E−06
	Worst	1.67E+06	7.13E+05	4.06E+06	1.59E+06	5.91E+00
	Ave	2.09E+05	1.56E+05	7.04E+05	3.11E+05	5.61E−01
	Std	2.20E+05	1.07E+05	6.97E+05	2.50E+05	9.58E−01

(continued on next page)

Table 6 (*continued*)

Function		SMO	WOA	GOA	PSO	SFS
F4	Best	6.55E+01	6.10E+01	7.58E+01	6.17E+01	5.85E−04
	Worst	9.44E+01	9.38E+01	9.98E+01	9.95E+01	6.90E−01
	Ave	8.20E+01	8.25E+01	9.53E+01	8.97E+01	1.37E−01
	Std	6.47E+00	7.51E+00	4.07E+00	6.91E+00	1.56E−01
F5	Best	5.43E+07	2.98E+07	1.58E+08	1.04E+08	2.87E+01
	Worst	3.49E+08	3.76E+08	6.61E+08	5.47E+08	3.40E+01
	Ave	1.81E+08	1.74E+08	4.15E+08	3.02E+08	2.92E+01
	Std	6.35E+07	6.65E+07	9.10E+07	1.07E+08	7.63E−01
F6	Best	2.87E+04	2.63E+04	5.53E+04	2.74E+04	3.70E+00
	Worst	8.14E+04	8.77E+04	1.22E+05	1.15E+05	6.43E+00
	Ave	5.33E+04	5.61E+04	8.87E+04	7.37E+04	5.09E+00
	Std	1.11E+04	1.16E+04	1.35E+04	1.90E+04	6.14E−01
F7	Best	1.51E+01	1.68E+01	6.84E+01	1.90E+01	4.16E−03
	Worst	1.74E+02	1.74E+02	3.63E+02	2.96E+02	1.20E−01
	Ave	6.78E+01	7.00E+01	2.15E+02	1.64E+02	4.02E−02
	Std	3.19E+01	2.99E+01	5.08E+01	6.38E+01	2.76E−02

Table 7
Results of multimodal functions.

Function		SMO	WOA	GOA	PSO	SFS
F8	Best	−6.40E+02	−1.71E+03	−6.03E+02	−1.66E+01	−9.75E+02
	Worst	−1.73E+01	−4.10E+02	−2.86E+00	−3.00E+00	−3.39E+02
	Ave	−2.77E+02	−1.22E+03	−2.77E+02	−9.58E+00	−6.32E+02
	Std	1.29E+02	3.12E+02	1.26E+02	3.02E+00	1.30E+02
F9	Best	0.00E+00	1.71E−13	7.51E+02	8.66E+01	0.00E+00
	Worst	0.00E+00	3.86E+02	1.43E+03	2.43E+02	0.00E+00
	Ave	0.00E+00	7.06E+01	1.12E+03	1.55E+02	0.00E+00
	Std	0.00E+00	8.99E+01	1.34E+02	3.46E+01	0.00E+00
F10	Best	8.88E−16	5.04E−09	2.09E+01	4.26E+00	1.00E+00
	Worst	8.88E−16	2.11E+01	2.16E+01	1.52E+01	1.00E+00
	Ave	8.88E−16	8.62E+00	2.14E+01	9.16E+00	1.00E+00
	Std	9.91E−32	8.66E+00	1.18E−01	2.56E+00	0.00E+00
F11	Best	0.00E+00	8.22E−10	1.02E+01	3.09E+00	0.00E+00
	Worst	0.00E+00	3.09E+00	2.69E+01	1.45E+01	0.00E+00
	Ave	0.00E+00	7.70E−01	1.64E+01	8.78E+00	0.00E+00
	Std	0.00E+00	5.73E−01	2.80E+00	2.52E+00	0.00E+00
F12	Best	5.77E−01	3.47E−01	2.17E+06	5.40E+01	6.69E−01
	Worst	1.67E+00	2.76E+06	6.52E+07	1.78E+07	1.61E+00
	Ave	1.55E+00	1.32E+05	2.25E+07	4.23E+06	1.26E+00
	Std	2.20E−01	4.02E+05	1.31E+07	4.08E+06	2.26E−01
F13	Best	2.99E+00	1.60E+00	6.06E+09	7.38E+08	2.61E+00
	Worst	3.00E+00	2.41E+09	4.51E+10	2.88E+10	3.00E+00
	Ave	3.00E+00	2.13E+08	2.07E+10	7.58E+09	2.98E+00
	Std	9.82E−04	5.12E+08	7.51E+09	5.05E+09	6.26E−02
Function		GWO	OIO	LCA	WDO	HS
F8	Best	−1.00E+03	−1.91E+03	−1.16E+03	−1.88E+03	−1.88E+03
	Worst	−2.67E+02	−1.64E+03	−5.00E+02	−2.21E+02	−1.69E+03
	Ave	−6.42E+02	−1.89E+03	−8.05E+02	−8.67E+02	−1.79E+03
	Std	1.48E+02	4.11E+01	1.33E+02	5.49E+02	3.49E+01
F9	Best	1.54E−02	3.54E−07	3.14E+02	4.94E−01	9.28E+01
	Worst	2.56E+02	3.25E+01	8.31E+02	2.75E+02	2.04E+02
	Ave	8.66E+01	1.43E+00	5.87E+02	2.00E+02	1.51E+02
	Std	5.86E+01	5.73E+00	1.01E+02	7.95E+01	2.20E+01
F10	Best	1.00E+00	1.00E+00	3.87E+00	1.00E+00	1.12E+00
	Worst	2.14E+00	1.00E+00	1.71E+01	1.10E+00	1.79E+00
	Ave	1.12E+00	1.00E+00	1.01E+01	1.02E+00	1.41E+00
	Std	2.00E−01	2.22E−06	2.31E+00	2.30E−02	1.34E−01
F11	Best	3.62E−02	1.41E−06	7.12E+00	5.70E−03	1.17E+00
	Worst	1.48E+00	2.04E−03	1.56E+01	1.11E+00	1.97E+00
	Ave	8.40E−01	2.07E−04	1.06E+01	7.58E−01	1.41E+00
	Std	3.55E−01	3.69E−04	1.99E+00	3.54E−01	1.32E−01
F12	Best	6.18E−01	2.30E−06	9.16E+05	7.77E−03	3.23E+00
	Worst	1.29E+02	1.00E−03	2.49E+07	4.54E+00	2.21E+01
	Ave	8.83E+00	1.75E−04	9.32E+06	1.11E+00	8.46E+00
	Std	1.53E+01	1.93E−04	6.09E+06	9.66E−01	2.95E+00
F13	Best	3.12E+00	2.13E−05	1.86E+09	3.24E−01	1.30E+06
	Worst	5.41E+08	6.44E−03	2.35E+10	9.00E+04	3.97E+07
	Ave	1.45E+07	8.79E−04	1.03E+10	1.95E+03	1.29E+07
	Std	6.32E+07	9.96E−04	4.30E+09	1.12E+04	8.77E+06

(*continued on next page*)

Table 7 (*continued*)

Function		SMO	WOA	GOA	PSO	SFS
Function		FA	ABC	BBO	BA	TLBO
F8	Best	−6.84E+02	−7.24E+02	−5.68E+02	−7.39E+02	−1.36E+03
	Worst	−4.31E+01	−1.03E+02	1.72E+02	8.96E+01	−2.76E+02
	Ave	−3.61E+02	−3.91E+02	−1.34E+02	−3.04E+02	−7.28E+02
	Std	1.37E+02	1.16E+02	1.66E+02	1.67E+02	2.39E+02
F9	Best	5.17E+02	5.44E+02	8.73E+02	5.39E+02	3.31E−06
	Worst	1.13E+03	1.03E+03	1.52E+03	1.51E+03	2.05E+01
	Ave	7.83E+02	7.64E+02	1.20E+03	1.08E+03	1.99E+00
	Std	1.27E+02	1.11E+02	1.32E+02	2.15E+02	3.54E+00
F10	Best	8.32E+00	8.43E+00	1.54E+01	9.12E+00	1.00E+00
	Worst	2.09E+01	2.21E+01	3.34E+01	2.82E+01	1.00E+00
	Ave	1.46E+01	1.45E+01	2.41E+01	1.85E+01	1.00E+00
	Std	3.00E+00	2.93E+00	3.47E+00	4.26E+00	4.18E−05
F11	Best	8.33E+00	9.73E+00	1.52E+01	7.64E+00	3.64E−08
	Worst	2.53E+01	2.38E+01	3.16E+01	2.81E+01	3.41E−01
	Ave	1.47E+01	1.54E+01	2.43E+01	1.95E+01	3.23E−02
	Std	3.13E+00	2.82E+00	3.38E+00	4.22E+00	6.93E−02
F12	Best	2.38E+06	6.32E+05	1.44E+07	1.55E+06	2.12E−01
	Worst	4.91E+07	4.09E+07	1.07E+08	8.16E+07	1.26E+00
	Ave	1.49E+07	1.59E+07	5.29E+07	3.19E+07	7.12E−01
	Std	9.05E+06	7.54E+06	1.67E+07	1.91E+07	2.57E−01
F13	Best	4.72E+09	5.11E+09	1.66E+10	5.56E+09	2.16E+00
	Worst	3.13E+10	3.08E+10	6.42E+10	5.37E+10	3.12E+00
	Ave	1.64E+10	1.69E+10	4.01E+10	2.87E+10	2.69E+00
	Std	6.21E+09	5.35E+09	1.00E+10	1.12E+10	1.76E−01

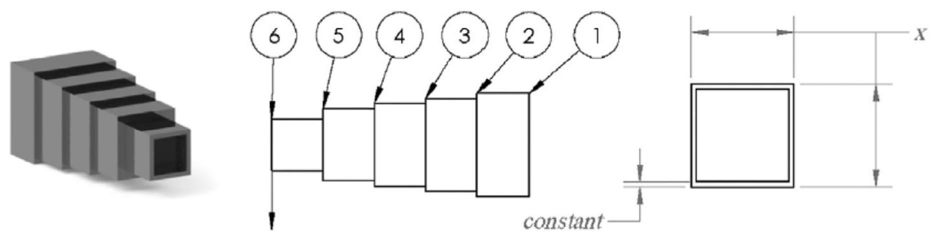

Fig 3. Cantilever beam design problem.

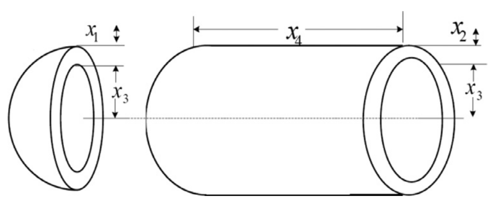

Fig 4. Pressure vessel design problem.

number of function evaluations compared to other methods. This algorithm has found the optimal solution with three search agents. CS is ranked the second with 166% increase in cumber of function evaluations.

4.3.2. Pressure vessel design problem

This problem is about designing a specific cylindrical tank with its ends blocked, using hemispheres. Like all optimization problems, purpose of this problem is to reduce cost of construction, material, formation and welding. Schematic of the problem is represented in Fig. 4. This problem has 4 parameters and 4 constraints which can be written as follows.

$$\text{Consider}\ \vec{X} = [x_1 x_2 x_3 x_4]$$

$$\text{Minimize}\ f(\vec{X}) = 0.62224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_4 x_1^2 + 19.84 x_3 x_1^2$$

$$\text{Subject to}\ g_1(\vec{X}) = -x_1 + 0.0193 x_3 \le 0$$

$$g_2(\vec{X}) = -x_3 + 0.00954 x_3 \le 0$$

$$g_3(\vec{X}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(\vec{X}) = x_4 - 240 \le 0$$

$$\text{Variable range}\ 0 \le x_1, x_2 \le 99$$

$$10 \le x_3, x_4 \le 200$$

In studies, different methods have been employed to solve this problem. some of these algorithms include VPL, charged system search (CSS) (Ali Kaveh & Siamak Talatahari, 2010a), Hybridizing particle swarm optimization with differential evolution (PSO-DE) (Liu, Cai, & Wang, 2010), galaxy-based search algorithm (GSA) (Shah-Hosseini, 2011), improved ACO (A. Kaveh & Siamak Talatahari, 2010b), co-evolutionary differential evolution (CDE) (Huang, Wang, & He, 2007), WOA, Evolution strategies (ES) (Mezura-Montes & Coello, 2008), different genetic algorithms (GA) (Coello, 2000; Coello & Montes, 2002; Deb, 1997), co-evolutionary particle swarm optimization approach (CPSO) (He & Wang, 2007), improved HS (Mahdavi, Fesanghary, & Damangir, 2007), augmented Lagrangian multiplier approach (ALM) (Kannan & Kramer, 1994), branch and bound method (Sandgren, 1990). Best result found by each method is given in Table 11. Satisfactory results of Table 11 show efficiency of SMO in solving this problem.

According to Table 11, SMO with 5 search agents and maximum number of 2500 function evaluations has obtained the optimal solution.

Table 8
Results of fixed-multimodal functions .

Function		SMO	WOA	GOA	PSO	SFS
F14	Best	9.99E−01	1.05E+00	2.08E+00	9.98E−01	9.98E−01
	Worst	1.27E+01	2.72E+02	5.00E+02	4.92E+02	1.27E+01
	Ave	1.18E+01	1.64E+01	2.93E+02	2.18E+01	1.05E+01
	Std	2.54E+00	2.81E+01	2.04E+02	6.32E+01	3.87E+00
F15	Best	8.51E−04	7.54E−04	1.68E−02	8.15E−04	4.54E−04
	Worst	1.48E−01	1.24E−01	3.61E+03	2.22E+01	1.09E−01
	Ave	8.93E−02	2.15E−02	7.90E+01	4.58E−01	2.55E−02
	Std	5.41E−02	2.53E−02	3.85E+02	2.42E+00	2.90E−02
F16	Best	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00
	Worst	−1.23E−01	1.76E+00	1.90E+03	1.42E−01	−2.10E−01
	Ave	−6.42E−01	−7.25E−01	3.36E+02	−6.73E−01	−9.37E−01
	Std	2.74E−01	4.95E−01	4.60E+02	4.14E−01	1.55E−01
F17	Best	3.99E−01	3.98E−01	6.52E−01	3.98E−01	3.98E−01
	Worst	5.56E+01	1.96E+01	1.54E+02	1.02E+02	9.21E+00
	Ave	1.99E+01	3.60E+00	2.58E+01	4.96E+00	6.51E−01
	Std	1.88E+01	4.77E+00	3.04E+01	1.47E+01	9.97E−01
F18	Best	3.02E+00	3.00E+00	4.01E+00	3.00E+00	3.00E+00
	Worst	6.00E+02	7.01E+03	8.54E+04	9.41E+02	9.27E+01
	Ave	2.10E+02	1.46E+02	5.26E+03	1.23E+02	2.11E+01
	Std	2.20E+02	7.29E+02	1.23E+04	2.62E+02	1.97E+01
F19	Best	−2.90E−01	−3.86E+00	−1.45E−01	−3.00E−01	−2.52E−01
	Worst	−9.74E−03	−6.22E−01	−6.79E−56	−1.47E−22	−1.01E−02
	Ave	−1.24E−01	−2.97E+00	−1.50E−03	−1.36E−01	−1.40E−01
	Std	6.72E−02	9.12E−01	1.45E−02	1.09E−01	5.30E−02
F20	Best	−2.75E+00	−2.39E+00	−2.18E−24	−1.53E+00	−2.84E+00
	Worst	0.00E+00	−5.11E−03	0.00E+00	0.00E+00	−3.14E−02
	Ave	−2.31E−01	−3.15E−01	−2.45E−26	−5.12E−02	−8.98E−01
	Std	4.06E−01	5.74E−01	2.18E−25	1.87E−01	6.58E−01
F21	Best	−8.88E+00	−9.45E+00	−1.18E+00	−1.02E+01	−8.00E+00
	Worst	−3.33E−01	−2.73E−01	−6.63E−02	−1.65E−01	−5.62E−01
	Ave	−1.67E+00	−1.58E+00	−2.16E−01	−3.11E+00	−3.15E+00
	Std	1.27E+00	1.56E+00	1.46E−01	2.90E+00	1.49E+00
F22	Best	−6.46E+00	−9.92E+00	−9.59E−01	−1.04E+01	−7.87E+00
	Worst	−6.07E−01	−2.94E−01	−1.14E−01	−2.76E−01	−7.53E−01
	Ave	−2.11E+00	−1.76E+00	−2.94E−01	−3.77E+00	−3.19E+00
	Std	1.27E+00	1.68E+00	1.26E−01	2.88E+00	1.52E+00
F23	Best	−7.57E+00	−7.51E+00	−1.06E+00	−1.05E+01	−1.04E+01
	Worst	−4.37E−01	−3.22E−01	−1.97E−01	−3.34E−01	−1.00E+00
	Ave	−2.17E+00	−1.76E+00	−4.13E−01	−3.13E+00	−3.54E+00
	Std	1.16E+00	1.43E+00	1.98E−01	2.55E+00	1.67E+00
Function		GWO	OIO	LCA	WDO	HS
F14	Best	1.59E+00	9.98E−01	9.98E−01	9.98E−01	9.98E−01
	Worst	5.77E+01	1.99E+00	5.00E+02	2.28E+01	5.93E+00
	Ave	1.23E+01	1.02E+00	1.04E+02	5.62E+00	1.19E+00
	Std	6.48E+00	1.40E−01	1.75E+02	6.24E+00	7.38E−01
F15	Best	3.34E−04	3.47E−04	9.85E−04	3.53E−04	4.12E−04
	Worst	1.49E−01	3.48E−03	8.67E+01	4.27E−02	6.33E−02
	Ave	2.49E−02	1.65E−03	1.03E+00	4.49E−03	1.03E−02
	Std	3.84E−02	7.13E−04	8.66E+00	6.37E−03	1.04E−02
F16	Best	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00
	Worst	−2.12E−01	−1.03E+00	1.80E+03	−7.54E−01	−1.03E+00
	Ave	−1.01E+00	−1.03E+00	5.02E+01	−1.02E+00	−1.03E+00
	Std	8.29E−02	3.33E−12	2.43E+02	3.43E−02	8.53E−05
F17	Best	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01
	Worst	2.07E+01	3.98E−01	1.17E+02	4.70E+00	3.98E−01
	Ave	3.72E+00	3.98E−01	8.38E+00	5.85E−01	3.98E−01
	Std	6.31E+00	8.93E−16	1.66E+01	5.59E−01	3.24E−05
F18	Best	3.00E+00	3.00E+00	3.00E+00	3.01E+00	3.00E+00
	Worst	6.11E+02	3.00E+00	5.04E+03	8.40E+02	9.71E+01
	Ave	5.08E+01	3.00E+00	3.34E+02	1.89E+01	1.89E+01
	Std	1.15E+02	7.38E−15	8.06E+02	8.38E+01	2.52E+01
F19	Best	−3.00E−01	−3.00E−01	−3.00E−01	−3.00E−01	−3.00E−01
	Worst	−3.00E−01	−3.00E−01	−1.42E−64	−1.02E−09	−3.00E−01
	Ave	−3.00E−01	−3.00E−01	−8.84E−02	−1.67E−01	−3.00E−01
	Std	5.02E−16	4.96E−05	1.17E−01	1.29E−01	5.02E−16
F20	Best	−3.31E+00	NA	−2.50E+00	−3.03E+00	−3.32E+00
	Worst	−4.38E−04	NA	0.00E+00	−8.81E−137	−3.20E+00
	Ave	−1.69E+00	NA	−2.15E−01	−3.81E−01	−3.27E+00
	Std	1.15E+00	NA	4.63E−01	6.88E−01	5.85E−02

(*continued on next page*)

Table 8 (*continued*)

Function		SMO	WOA	GOA	PSO	SFS
F21	Best	−1.00E+01	−1.02E+01	−1.00E+01	−1.01E+01	−1.02E+01
	Worst	−3.55E−01	−5.56E+00	−2.10E−01	−1.11E+00	−2.63E+00
	Ave	−4.35E+00	−1.01E+01	−2.45E+00	−6.34E+00	−5.77E+00
	Std	3.25E+00	5.49E−01	2.61E+00	3.05E+00	3.24E+00
F22	Best	−1.03E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	Worst	−3.54E−01	−3.72E+00	−1.69E−01	−1.42E+00	−1.84E+00
	Ave	−3.99E+00	−1.03E+01	−2.88E+00	−6.61E+00	−5.04E+00
	Std	3.14E+00	6.68E−01	2.54E+00	3.08E+00	3.15E+00
F23	Best	−1.03E+01	−1.05E+01	−1.04E+01	−1.05E+01	−1.05E+01
	Worst	−5.57E−01	−1.05E+01	−3.39E−01	−1.41E+00	−1.68E+00
	Ave	−4.15E+00	−1.05E+01	−2.59E+00	−6.69E+00	−4.42E+00
	Std	3.07E+00	7.07E−03	2.02E+00	3.37E+00	3.08E+00
Function		FA	ABC	BBO	BA	TLBO
F14	Best	9.98E−01	9.98E−01	3.43E+00	1.62E+00	9.98E−01
	Worst	5.00E+02	5.00E+02	5.00E+02	5.00E+02	1.83E+01
	Ave	3.58E+01	2.13E+01	4.65E+02	2.91E+02	9.47E+00
	Std	9.61E+01	5.51E+01	9.60E+01	2.12E+02	3.62E+00
F15	Best	8.82E−04	1.54E−03	3.07E−02	1.23E−03	4.49E−04
	Worst	9.22E+01	4.16E+01	1.33E+03	4.52E+03	1.40E−01
	Ave	1.84E+00	1.08E+00	5.13E+01	5.25E+01	3.26E−02
	Std	9.94E+00	4.58E+00	1.59E+02	4.53E+02	3.78E−02
F16	Best	−1.03E+00	−1.03E+00	−8.46E−01	−1.03E+00	−1.03E+00
	Worst	3.98E+01	1.34E+02	3.18E+03	1.13E+03	−4.57E−09
	Ave	4.48E−01	1.88E+00	4.44E+02	4.62E+01	−7.58E−01
	Std	5.60E+00	1.47E+01	6.69E+02	1.52E+02	3.67E−01
F17	Best	3.98E−01	3.98E−01	7.40E−01	4.16E−01	3.98E−01
	Worst	1.80E+02	7.92E+01	2.19E+02	1.31E+02	1.78E+01
	Ave	1.08E+01	6.85E+00	4.55E+01	1.58E+01	1.76E+00
	Std	2.14E+01	1.37E+01	4.96E+01	2.32E+01	2.35E+00
F18	Best	3.00E+00	3.00E+00	5.00E+00	3.00E+00	3.00E+00
	Worst	2.02E+03	2.77E+03	1.43E+05	1.20E+04	2.35E+02
	Ave	1.22E+02	1.21E+02	1.12E+04	7.08E+02	3.97E+01
	Std	3.18E+02	3.60E+02	2.35E+04	1.84E+03	3.64E+01
F19	Best	−3.00E−01	−2.95E−01	−5.22E−02	−3.00E−01	−3.00E−01
	Worst	−1.14E−27	−9.80E−30	−4.96E−78	−2.25E−29	−3.00E−01
	Ave	−7.87E−02	−7.81E−02	−6.97E−04	−1.70E−01	−3.00E−01
	Std	1.09E−01	1.03E−01	5.40E−03	1.24E−01	5.02E−16
F20	Best	−7.14E−14	−2.62E−07	−2.92E−20	−2.00E−01	−3.29E+00
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	−5.16E−02
	Ave	−7.14E−16	−2.62E−09	−3.15E−22	−2.83E−03	−1.66E+00
	Std	7.14E−15	2.62E−08	2.92E−21	2.03E−02	1.13E+00
F21	Best	−1.02E+01	−1.02E+01	−8.85E−01	−2.49E+00	−9.98E+00
	Worst	−1.59E−01	−1.26E−01	−8.14E−02	−9.47E−02	−4.93E−01
	Ave	−1.21E+00	−1.58E+00	−2.14E−01	−4.30E−01	−2.94E+00
	Std	1.71E+00	2.25E+00	1.29E−01	3.88E−01	2.02E+00
F22	Best	−1.04E+01	−1.04E+01	−6.92E−01	−2.32E+00	−5.73E+00
	Worst	−2.14E−01	−1.71E−01	−1.02E−01	−1.40E−01	−6.00E−01
	Ave	−2.09E+00	−1.67E+00	−2.68E−01	−4.74E−01	−2.64E+00
	Std	2.43E+00	1.82E+00	1.27E−01	3.43E−01	1.43E+00
F23	Best	−1.05E+01	−1.05E+01	−1.46E+00	−5.47E+00	−8.62E+00
	Worst	−2.79E−01	−3.24E−01	−1.83E−01	−2.19E−01	−5.22E−01
	Ave	−2.36E+00	−2.31E+00	−4.11E−01	−7.03E−01	−2.90E+00
	Std	2.41E+00	2.12E+00	2.20E−01	6.47E−01	1.67E+00

Table 9
Summarized results of Tables 6–8.

SMO vs#	WOA	GOA	PSO	SFS	GWO	OIO	LCA	WDO	HS	FA	ABC	BBO	BA	TLBO
Better	13	22	15	11	14	12	17	12	15	17	17	22	21	12
As good as	0	0	0	2	0	0	0	0	0	0	0	0	0	0
Worst	10	1	8	10	9	11	6	11	8	6	6	1	2	11

Table 10
Comparison results for cantilever design problem.

Algorithm	Optimal values					Optimal cost	No. evaluation
	x_1	x_2	x_3	x_4	x_5		
SMO	5.78691656	5.051692965	4.227690161	3.73905533	2.257392706	1.31095	1500
BBO	5.755852701	4.589896046	5.015255584	3.398046751	2.436181023	1.31919	25,000
LCA	6.03502832	5.414056302	4.288538271	3.329424921	2.017602678	1.31231	45,000
ALO	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995	14,000
SSA	6.015135	5.309305	4.495007	3.501426	2.152788	1.33996	N.A.
GOA	6.011674	5.31297	4.48307	3.50279	2.16333	1.33996	13,000
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996	15,000
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999	2500
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	N.A.
GCA_I	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	N.A.
GCA_II	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	N.A.

Table 11
Comparison results for pressure vessel design problem.

Algorithm	Optimal values				Optimal cost	No. evaluation
	T_s	T_h	R	L		
SMO	0.824248041	0.407150024	42.36584829	173.7973031	6021.6291	2500
VPL	0.815200	0.426500	42.0912541	176.742314	6044.9565	N.A.
GWO	0812500	0434500	42.089181	176.758731	6051.5639	N.A.
CSS	0.812500	0.437500	42.103624	176.572656	6059.0888	N.A.
PSO-DE	0.8125	0.4375	42.098445596	176.636596	6059.7143	42,100
OIO	N.A.	N.A.	N.A.	N.A.	6059.7143	50,000
SFS	0.8125	0.4375	42.0984456	176.636596	6059.7143	50,000
ACO	0.812500	0.437500	42.098353	176.637751	6059.7258	N.A.
CDE	0.812500	0.437500	42.098411	176.637690	6059.7340	204,800
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410	6300
ES	0.812500	0.437500	42.098087	176.640518	6059.7456	25,000
GA (Coello & Montes)	0.937500	0.437500	42.097398	176.654050	6059.9463	N.A.
PSO (He & Wang)	0.812500	0.437500	42.091266	176.746500	6061.0777	14,790
GA (Coello)	0.812500	0.437500	40.323900	200.000000	6288.7445	N.A.
GA (Deb & Gene)	0.812500	0.437500	48.329000	112.679000	6410.3811	N.A.
BA (Gandomi, Yang, Alavi, & Talatahari, 2013)	0.8125	0.4375	42.098446	176.636596	6059.7143	15,000
Improved HS	1.125000	0.625000	58.29015	43.69268	7197.730	N.A.
ALM	1.125000	0.625000	58.2910	43.69	7198.0428	N.A.
HS (Chauhan, Patel, & Arekar, 2017)	1.125	0.625	58.219	44.037	7200.202	N.A.
Branch and Bound	1.125000	0.625000	47.7000	117.7010	8129.1036	N.A.
GSA	1.125000	0.625000	55.988659	84.4542023	8538.8359	7110

5. Conclusion

In this paper, a novel swarm-based optimization algorithm inspired by mimicking behavior of people is proposed. The proposed method (called SMO) includes a mimic operator to simulate search in the response space of the optimization problem. SMO is very simple because it does not require control operator. On the other hand, the idea proposed to perform exploration and exploitation provides the ability for SMO to overcome general drawback of the population-based methods which requires higher number of initial population. In order to evaluate efficiency of SMO, a set of 23 benchmark functions and two classic engineering problems are selected to analyze exploration, exploitation, avoiding local optima, convergence behavior and ability to solve problem with high number of decision variables. Next, SMO is compared with 14 well-known and state of the art optimization algorithms. Results indicate higher efficiency of SMO compared to the mentioned algorithms.

The most important feature of SMO is that it does not require control parameters. Accordingly, the proposed method does not require a time-consuming process to adjust parameters like most of the presented methods. On the other hand, simple imitation concept reduces number of initial population to less than 5 search agents. This characteristic reduces number of function evaluations as a result of which speed of SMO increases significantly making it suitable for solving problem with high-dimensional decision variable and complicated optimality function. Simple understanding and coding are other features of SMO. According to results of Fig. 2, this method converges to the optimal response in less than 10 iterations.

Future studies on SMO algorithm are divided to two classes including algorithm development and real world application. Developing the binary and multi-objective version of the algorithm is possible for solving discrete problems. Another issue which can be studied about this algorithm is selection of proper number of initial population for different problems. Although this adds to control parameters of the proposed algorithm, but it can improve speed of exploration and exploitation in different problems. In this algorithm, number of agents (initial population) is considered as the control parameter in all iterations. By changing number of search agents in different steps of exploration and exploitation, more control can be applied. Solving optimization problems in different contexts might be a valuable development of the proposed algorithm.

Conflict of interest

None.

Credit authorship contribution statement

Saeed Balochian: Conceptualization, Data curation, Formal analysis, Methodology, Writing - original draft, Writing - review

& editing. **Hossein Baloochian:** Conceptualization, Data curation, Formal analysis, Methodology, Writing - original draft, Writing - review & editing.

References

Abbass, H. A. (2001). MBO: Marriage in honey bees optimization-a Haplometrosis polygynous swarming approach. *Paper presented at the proceedings of the 2001 congress on evolutionary computation*. IEEE Cat. No.01TH8546.

Adrian, A., Utamima, A., & Wang, K.-J. (2014). A comparative study of GA, PSO and ACO for solving construction site layout optimization (Vol. 19).

Bayraktar, Z., Komurcu, M., & Werner, D. H. (2010). Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. *The 2010 IEEE antennas and propagation society international symposium*.

Bekiroğlu, S., Dede, T., & Ayvaz, Y. (2009). Implementation of different encoding types on structural optimization based on adaptive genetic algorithm. *Finite Elements in Analysis and Design, 45*(11), 826–835. hTtps://doi.org/10.1016/j.finel.2009.06.019.

Bhambu, P., Sharma, S., & Kumar, S. (2018). Modified Gbest artificial bee colony algorithm, Singapore.

Chauhan, G., Patel, V., & Arekar, V. (2017). Applications of harmony search algorithm. *Kalpa Publications in Civil Engineering, 1*(2017), 239–247.

Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures, 139*, 98–112.

Chickerman, H., & Gea, H. C. (1996). Structural optimization using local approximation method. *International Journal for Numerical Methods in Engineering, 39*(5), 829–846 dOi:10.1002/(SICI)1097-0207(19960315)39:5<829::AID-NME884>3.0.CO;2-U.

Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*(2), 113–127. hTtps://doi.org/10.1016/S0166-3615. 99)00046-9.

Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*(3), 193–203. hTtps://doi.org/10.1016/S1474-0346. 02)00011-3.

Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications, 40*(16), 6374–6384. hTtps://doi.org/10.1016/j.eswa.2013.05.041.

Davis, L. (1991). *Bit-climbing, representational bias, and test suite design*. Paper presented at the ICGA.

Deb, K. (1997). GeneAS: A robust optimal design technique for mechanical component design. In D. Dasgupta, & Z. Michalewicz (Eds.), *Evolutionary Algorithms in engineering applications* (pp. 497–514). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dede, T., & Togan, V. (2015). A teaching learning based optimization for truss structures with frequency constraints. [A teaching learning based optimization for truss structures with frequency constraints]. *Structural Engineering and Mechanics, 53*(4), 833–845. doi:10.12989/SEM.2015.53.4.833.

Dorigo, M. (1992). *Optimization, learning and natural algorithms* PhD Thesis. Politecnico di Milano.

Fadakar, E., & Ebrahimi, M. (2016). A new metaheuristic football game inspired algorithm. *The 2016 1st conference on swarm intelligence and evolutionary computation (CSIEC)*.

Fister, I., Fister, I., Yang, X.-S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation, 13*, 34–46. hTtps://doi.org/10.1016/j.swevo.2013.06.001.

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(1), 17–35. doi:10.1007/s00366-011-0241-y.

Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications, 22*(6), 1239–1255. doi:10.1007/s00521-012-1028-9.

Gao, X. Z., Govindasamy, V., Xu, H., Wang, X., & Zenger, K. (2015). Harmony search method: Theory and applications. *Computational Intelligence and Neuroscience, 2015*, 10. doi:10.1155/2015/258491.

Gong, C. (2016). Opposition-based adaptive fireworks algorithm. *Algorithms, 9*(3), 1–13. doi:10.3390/a9030043.

Gong, M., Wang, S., Liu, W., Yan, J., & Jiao, L. (2016). Evolutionary computation in China: A literature survey. *CAAI Transactions on Intelligence Technology, 1*(4), 334–354.

He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*(1), 89–99. hTtps://doi.org/10.1016/j.engappai.2006.03.003.

Huang, F.-z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186*(1), 340–356. hTtps://doi.org/10.1016/j.amc.2006.07.105.

Husseinzadeh Kashan, A. (2014). League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing, 16*, 171–200. hTtps://doi.org/10.1016/j.asoc.2013.12.005.

Husseinzadeh Kashan, A. (2015). A new metaheuristic for optimization: Optics inspired optimization (OIO). *Computers & Operations Research, 55*, 99–125. hTtps://doi.org/10.1016/j.cor.2014.10.011.

Javidy, B., Hatamlou, A., & Mirjalili, S. (2015). Ions motion algorithm for solving optimization problems. *Applied Soft Computing, 32*, 72–79. hTtps://doi.org/10.1016/j.asoc.2015.03.035.

Joyce, T., & Herrmann, J. M. (2018). A review of no free lunch theorems, and their implications for metaheuristic optimisation. In X.-S. Yang (Ed.), *Nature-inspired algorithms and applied optimization* (pp. 27–51). Cham: Springer International Publishing.

Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*(2), 405–411. doi:10.1115/1.2919393.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471. doi:10.1007/s10898-007-9149-x.

Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software, 59*, 53–70. hTtps://doi.org/10.1016/j.advengsoft.2013.03.004.

Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations, 27*(1), 155–182. doi:10.1108/02644401011008577.

Kaveh, A., & Talatahari, S. (2010). Optimal design of skeletal structures via the charged system search algorithm. *Structural and Multidisciplinary Optimization, 41*(6), 893–911. doi:10.1007/s00158-009-0462-5.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Krishnanand, K. N., & Ghose, D. (2005). Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. *The proceedings 2005 IEEE swarm intelligence symposium*: 2005. SIS 2005.

Lacoste-Badie, S., & Droulers, O. (2014). *Advertising memory: The power of mirror neurons* (Vol. *7*).

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*(2), 629–640. hTtps://doi.org/10.1016/j.asoc.2009.08.031.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*(2), 1567–1579. hTtps://doi.org/10.1016/j.amc.2006.11.033.

Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems, 149*, 153–165. hTtps://doi.org/10.1016/j.chemolab.2015.08.020.

Meryem, B., & Abdelmadjid, B. (2016). Quantum inspired algorithm for a VRP with heterogeneous fleet mixed backhauls and time windows. *International Journal of Applied Metaheuristic Computing (IJAMC), 7*(4), 18–38. doi:10.4018/IJAMC.2016100102.

Mezura-Montes, E., & Coello, C. A. C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems, 37*(4), 443–473. doi:10.1080/03081070701303470.

Mirjalili, S. (2015a). The ant lion optimizer. *Advances in Engineering Software, 83*, 80–98. hTtps://doi.org/10.1016/j.advengsoft.2015.01.010.

Mirjalili, S. (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249. hTtps://doi.org/10.1016/j.knosys.2015.07.006.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.

Mirjalili, S., Gandomi, A., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. doi:10.1016/j.advengsoft.2017.07.002.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1–14. hTtps://doi.org/10.1016/j.swevo.2012.09.002.

Mirjalili, S., & Lewisa, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. hTtps://doi.org/10.1016/j.advengsoft.2013.12.007.

Moghdani, R., & Salimifard, K. (2018). Volleyball premier league algorithm. *Applied Soft Computing, 64*, 161–185. hTtps://doi.org/10.1016/j.asoc.2017.11.043.

Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: A survey. *International Journal of Bio-Inspired Computation, 3*(1), 1–16. doi:10.1504/ijbic.2011.038700.

Rao, R. V., & Patel, V. (2013). An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica, 20*(3), 710–720. hTtps://doi.org/10.1016/j.scient.2012.12.005.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303–315. hTtps://doi.org/10.1016/j.cad.2010.12.015.

Rizzolatti, G., & Fabbri-Destro, M. (2010). Mirror neurons: From discovery to autism. *Experimental Brain Research, 200*(3), 223–237. doi:10.1007/s00221-009-2002-3.

Salimi, H. (2015). Stochastic fractal search: A powerful metaheuristic algorithm. *Knowledge-Based Systems, 75*, 1–18. hTtps://doi.org/10.1016/j.knosys.2014.07.025.

Sandgren, E. (1990). Nonlinear Integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design, 112*(2), 223–229. doi:10.1115/1.2912596.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software, 105*, 30–47. hTtps://doi.org/10.1016/j.advengsoft.2017.01.004.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation (Vol. 6).

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*(6), 702–713. doi:10.1109/TEVC.2008.919004.

Tan, W. S., Hassan, M. Y., Shah Majid, M., & Abdul Rahman, H. (2013). *Optimal distributed renewable generation planning: A review of different approaches* (Vol. 18).

Tilahun, S. L., & Ong, H. C. (2012). Modified firefly algorithm. *Journal of Applied Mathematics, 2012*, 12. doi:10.1155/2012/467631.

Topal, A. O., & Altun, O. (2016). A novel meta-heuristic algorithm: Dynamic virtual bats algorithm. *Information Sciences, 354*, 222–235.

Topal, U., Dede, T., & Öztürk, H. T. (2017). Stacking sequence optimization for maximum fundamental frequency of simply supported antisymmetric laminated composite plates using Teaching–learning-based Optimization. *KSCE Journal of Civil Engineering, 21*(6), 2281–2288. doi:10.1007/s12205-017-0076-1.

Wang, G.-G., Deb, S., & Cui, Z. (2015). Monarch butterfly optimization. *Neural Computing and Applications*, 1–20. doi:10.1007/s00521-015-1923-y.

Wang, G. G., Deb, S., & Coelho, L. d. S. (2015). Elephant herding optimization. *The 2015 3rd international symposium on computational and business intelligence (ISCBI).*

Yang, X.-S. (2010a). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Yang, X.-S. (2010b). A New metaheuristic bat-inspired algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yu, J. J. Q., & Li, V. O. K. (2015). A social spider algorithm for global optimization. *Applied Soft Computing, 30*, 614–627. hTtps://doi.org/10.1016/j.asoc.2015.02.014.

Zhao, R., & Tang, W. (2008). *Monkey algorithm for global numerical optimization* (Vol. *2*).