Periodica Polytechnica Civil Engineering, 66(1), pp. 96–111, 2022

A Novel Hybrid Particle Swarm Optimization and Sine Cosine Algorithm for Seismic Optimization of Retaining Structures

Mohammad Khajehzadeh[1*], Alireza Sobhani[2], Seyed Mehdi Seyed Alizadeh[3], Mahdiyeh Eslami[4]

[1] Department of Civil Engineering, Anar Branch, Islamic Azad University, 7741943615, Anar, Iran
[2] Department of Petroleum Engineering, Tehran Science and Research Branch, Islamic Azad University, 1477893855, Tehran, Iran
[3] Petroleum Engineering Department, Australian College of Kuwait, 40005, West Mishref, Kuwait
[4] Department of Electrical Engineering, Kerman Branch, Islamic Azad University, 7635131167, Kerman, Iran
[*] Corresponding author, e-mail: mohammad.khajehzadeh@anariau.ac.ir

Received: 31 July 2021, Accepted: 20 September 2021, Published online: 29 September 2021

Abstract

This study introduces an effective hybrid optimization algorithm, namely Particle Swarm Sine Cosine Algorithm (PSSCA) for numerical function optimization and automating optimum design of retaining structures under seismic loads. The new algorithm employs the dynamic behavior of sine and cosine functions in the velocity updating operation of particle swarm optimization (PSO) to achieve faster convergence and better accuracy of final solution without getting trapped in local minima. The proposed algorithm is tested over a set of 16 benchmark functions and the results are compared with other well-known algorithms in the field of optimization. For seismic optimization of retaining structure, Mononobe-Okabe method is employed for dynamic loading condition and total construction cost of the structure is considered as the objective function. Finally, optimization of two retaining structures under static and seismic loading are considered from the literature. As results demonstrate, the PSSCA is superior and it could generate better optimal solutions compared with other competitive algorithms.

Keywords

retaining structure, seismic load, particle swarm, hybrid algorithm

1 Introduction

Many real world design problems can be considered as optimization problems and appropriate optimization method are required for the solution. On the other hand, the design problems have become more complicated when discontinuities, incomplete information, dynamicity, and uncertainties are involved. In such a case, classical optimization algorithms based on the mathematical principles demand exponential time or may not find the optimal solution at all. To overcome the mentioned problem, during the last few decades, introducing new efficient metaheuristic optimization algorithms to deal with the drawbacks of classical techniques have been of great concern. The privileges of these algorithms include derivation-free mechanisms, simple concepts and structure, local optima avoidance and effective for discrete and continuous functions. Accordingly, there is an increasing interest in presenting new metaheuristic algorithms, which offer higher accuracy and efficiency in dealing with complex optimization problems.

Generally, metaheuristic algorithms are of two types: single solution based methods and population based algorithms. As the name indicates, in the former type, only one solution is generated (usually at random) and processed during the optimization phase until a stopping criterion is satisfied. Some of these methods are Simulated Annealing [1], Tabu Search [2], Iterated Local Search [3] and Vortex Search Algorithm [4]. In the latter type, a set of solutions (i.e., population) is generated randomly and updated iteratively in each iteration of the optimization process until satisfying stopping criteria. Some well-known examples of these algorithms are the Genetic Algorithm [5], Ant Colony Optimization [6], Particle Swarm Optimization [7], Harmony search [8], and Harris hawks optimization [9].

Although all population-based search techniques may provide relatively satisfactory results, there is no metaheuristic algorithm providing a superior performance than others in solving all optimizing problems. In other words,

Cite this article as: Khajehzadeh, M., Sobhani, A., Alizadeh, S. M. S., Eslami, M. "A Novel Hybrid Particle Swarm Optimization and Sine Cosine Algorithm for Seismic Optimization of Retaining Structures", Periodica Polytechnica Civil Engineering, 66(1), pp. 96–111, 2022. https://doi.org/10.3311/PPci.19027

an algorithm may solve some problems better and some problems worse than the others [10]. Therefore, several studies have been undertaken to propose a novel algorithm or improve the performance and efficiency of the exiting metaheuristics [7, 11–14]. In the current research, a new hybrid optimization technique based on Particle Swarm Optimization (PSO) and Sine Cosine Algorithm (SCA) is developed. PSO is one of the most practical optimization algorithm which, has a simple structure and can be easily applied [15]. The proposed hybrid algorithm employs the advantages of sine and cosine functions in the velocity updating formula of the standard PSO algorithm. The proposed particle swarm sine cosine algorithm (PSSCA) utilizes a new weighting function as well as oscillation behavior of the sine and cosine mathematical functions which, can significantly improves the performance and provide a well balance between exploration and exploitation of the algorithm.

Reinforced concrete cantilever retaining structures are widely used in the field of civil engineering and frequently constructed for a variety of applications. Traditionally, in the design of retaining structures, initial assumed dimensions will be checked for stability and other building code requirements. If the dimensions could not satisfy the constraints, they would change repeatedly until satisfying all the requirements. In addition, in this time-consuming iterative process, the construction cost is not considered. In the optimum design of retaining structures, the dimensions, which provide minimum cost or weight of the structure and satisfy all the requirements, are defined automatically. Optimum design of these structures is a difficult optimization problem especially in case of seismic loading condition. However, in the earthquake-prone zone the design of the retaining walls under seismic loading should be strongly considered. There are numerous studies on the optimization of retaining structures under static loads [16–20]. However, the research into the optimum design of these structures under seismic loading is limited [21–24]. Due to the effectiveness of the proposed PSSCA, the applicability of this method for solving difficult optimization problems will be investigated via seismic optimization of retaining structures.

2 Particle Swarm Optimization (PSO)

PSO is a population-based optimization technique introduced by Kennedy and Eberhart [7]. In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each solution called a particle, flies in the problem search space looking for the optimal position to land. A particle, during the generations, adjusts its position according to its own experience as well as the experience of neighboring particles. A particle status on the search space is characterized by two factors: its position (x_i) and velocity (v_i). The new position and velocity of particles will be updated according to the following equations [20]:

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{1}$$

$$v_i^{t+1} = w.v_i^t + C_1 rand_1.\left[pbest_i - x_i^t\right] + C_2 rand_2.\left[gbest_i - x_i^t\right], \tag{2}$$

where, v_i^t is the velocity of particle i at iteration t, x_i^t represents the position of particle i, w is a weighting function, $pbest$ represents the best previous position of particle i, $gbest$ is the best solution so far, $rand_1$ and $rand_2$ are two independently uniformly distributed random number between 0 and 1, C_1 and C_2 are acceleration coefficients. The weighting function w will be obtained using the following equation:

$$w = w_{\max} - (w_{\max} - w_{\min}) \times t / t_{\max}, \tag{3}$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum values of w.

3 Sine Cosine Algorithm (SCA)

SCA is one of the recently developed population-based meta-heuristic method based on the mathematical features of sine and cosine functions [25]. In this algorithm, after generating the random initial solutions, each solution dynamically updates the positions according to the following equations:

$$\begin{cases} x_i^{t+1} = x_i^t + A \times \sin(r_1) \times \left| r_2 \times x_{Best} - x_i^t \right| & \text{if } r_3 < 0.5 \\ x_i^{t+1} = x_i^t + A \times \cos(r_1) \times \left| r_2 \times x_{Best} - x_i^t \right| & \text{otherwise} \end{cases}, \tag{4}$$

where, x_i^t represents the position of ith solution at iteration t, x_{Best} is the best solution in the population, r_1 is a random numbers in the range of [0, 2π], r_2 is a random weight of the best solution in the range of [–2, 2], r_3 is a random number between 0 and 1, and the symbol | . | represents absolute value. If the parameter r_3 is smaller than 0.5, the candidate solution chooses the sine function to update its position. The parameter A is a function to help the balance between exploration and exploitation of a search space and may be defined as follows:

$$A = 2 - 2\left(\frac{t}{t_{\max}}\right) \tag{5}$$

4 Hybrid PSSCA

In the proposed hybrid algorithm, the candidate solutions (i.e., particles) update their positions using the velocity parameter of the PSO algorithm. However, instead of simple random values in Eq. (2) ($rand_1$ and $rand_2$), the PSSCA utilizes sine and cosine functions which, successfully applied in the SCA [25]. The oscillation behavior of sine and cosine functions allows one solution to be re-positioned around another one and it can guarantee exploitation of the space defined between two solutions. In addition, the exploration of the algorithm will be modified by increasing the range of sine and cosine functions, which allow a solution to update its position outside the space between itself and another solution. To further improvement of the algorithm, the weighting function (w) of Eq. (2) will be replaced by a decreasing exponential function to control the balance between global search in early iterations and local search in late iterations.

The proposed PSSCA starts the search process with initial random candidate solutions (swarm of particles). In every iteration, the algorithm updates the position of the particles using a velocity parameter until satisfying some termination criteria. The detailed mathematical expression of PSSCA is presented in Section 4.1.

4.1 Algorithmic steps

Mathematically, the PSSCA algorithm has three main parts including population initialization, population evaluation, and updating the current population. Step-by-step procedure of the proposed PSSCA is presented as follows.

Step 1 population initialization

PSSCA starts the search process with a set of randomly generated particles (possible solutions) in the search space according to the following equation:

$$x_i = lb_i + rand \times \left(ub_i - lb_i\right); \;\; i = 1, 2, \ldots, N \, , \tag{6}$$

where x_i presents the location of ith particle in the search space. Moreover, ub_i and lb_i are the lower and upper bounds of the solution, respectively.

Step 2 population evaluation

In this step, initial population will be evaluated based on the objective function and the object with the best fitness value selected as $gbest_i$.

Step 3 golden change

In the third step, the particles will be sorted according to their fitness and the particle with the worst fitness will be changed by a random solution.

Step 4 velocity evaluation

In each iteration of optimization process, the particles are moved toward the best solution using velocity parameter (v_i). In the first iteration of optimization process, v_i will be generated randomly according to the following equation:

$$v_i\left(1\right) = randn^2 \, , \tag{7}$$

where $randn$ is a normally distributed pseudorandom number (obtained using $randn$ function in MATLAB). During the iterations, v_i will be updated using Eq. (8).

$$\begin{aligned} v_i^{t+1} &= w.v_i^t + C.\cos(rand_1).\left[pbest_i - x_i^t\right] \\ &+ C.\sin(rand_2).\left[gbest_i - x_i^t\right] \end{aligned} \tag{8}$$

In Eq. (8), C is a random number between 0 and 2 and functions sine and cosine take arguments in radians. In order to improve the search performance and controlling the balance between global search in early iterations and local search in late iterations, w will be evaluated by:

$$w = 100 \times \exp\left(-20 \times \frac{t}{t_{\max}}\right), \tag{9}$$

where $t_{\max}$ is the maximum number of iterations.

Step 5 velocity limitation

In order to clamp the particles movement, a reasonable interval is applied according to:

$$-v_{i\max} \le v_i \le v_{i\max} \, , \tag{10}$$

where, $v_{i\max}$ is a maximum movement allowed based on the following equation:

$$v_{i\max} = 0.1 \times \left(ub_i - lb_i\right). \tag{11}$$

Step 6 update position (generate new population)

In this stage, the particles move toward the global optimum in the search space based on Eq. (1).

The pseudo code of the proposed PSSCA is presented in Algorithm 1.

5 Seismic analysis of retaining structures

One of the important problems of structural engineering is seismic analysis of a retaining structure, especially in seismic zones. However, evaluation of accurate behavior of these structures will be more complicated while seismic loads are applied. Therefore, an effective pseudo-static approach will be applied to determine the real behavior of the structure under seismic loads. The first step in the analysis of retaining structures is evaluation of active and

Algorithm 1 The pseudo code of PSSCA algorithm

```
Determine the parameters N, t_max
Generate initial population using Eq. (6)
Generate initial velocity randomly
Calculate v_max from Eq. (11)
    t = 1
        while t < t_max                    //particles' movement
            Evaluate particles' fitness
            Update pbest_i and gbest_i
            Change the worst particle with a random one
            Determine w from Eq. (9)
            Calculate v_i using Eq. (8)
            Check velocity limitation
            Update particles' position based on Wq. (1)
          t = t + 1
        end while
Output the best solution
```

passive earth pressure acting on a wall. One of the most commonly used pseudo-static approach for calculating the distribution of seismic earth pressure is Mononobe-Okabe (M-O) method [26–29]. Fig. 1 depicted general forces acting on one-meter length of retaining structure. In this figure, P_{AE} and P_{PE} are the active and passive earth pressure under seismic loading, respectively. H is total height of the wall; β is the backfill slope angle; D is the depth of soil in front of the wall; q is the distributed surcharge load; $q_{\max}$ and q_{min} are the maximum and minimum contact pressure.

According to the M-O theory, a total active earth force can be evaluated based on the following expression [26]:

$$P_{AE} = \frac{1}{2}\gamma H^2 \left(1 - K_V\right) K_{AE} . \tag{12}$$

In Eq. (12), K_V is the vertical acceleration coefficient and K_{AE} is the dynamic active earth pressure coefficient defined as:

$$K_{AE} = \frac{\sin^2\left(\varnothing + \alpha - \theta\right)}{\cos(\theta)\sin^2(\alpha)\sin(\alpha - \delta - \theta)\left[1 + \sqrt{\frac{\sin(\delta + \varnothing)\sin(\varnothing - \theta - \beta)}{\sin(\alpha - \delta - \theta)\sin(\alpha + \beta)}}\right]^2} \tag{13}$$

where, α is angle of the back face of the wall and θ is the seismic inertia angle based on the following equation:

$$\theta = \tan^{-1}\left(\frac{K_h}{1 - K_V}\right), \tag{14}$$

where, K_h and K_V are the horizontal and vertical acceleration coefficients respectively, and can be defined as follows:

$$K_h = \frac{\text{horizontal earthquake acceleration component}}{\text{acceleration due to gravity}(g)}, \tag{15}$$

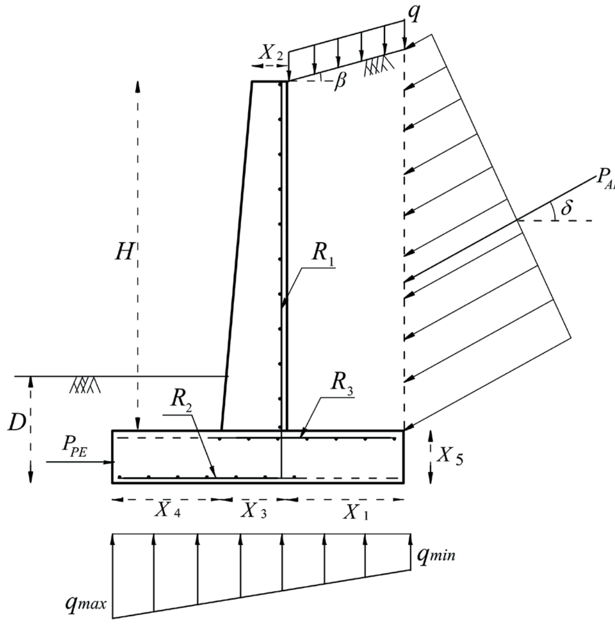

Fig. 1 Cross section of retaining structure

$$K_V = \frac{\text{vertical earthquake acceleration component}}{\text{acceleration due to gravity}(g)} . \tag{16}$$

It should be noted that, the acting point of P_{AE} ($\bar{y}$), can be computed utilizing Eq. (17)

$$\bar{y} = \frac{P_A\left(H/3\right) + \Delta P_{AE}\left(0.6H\right)}{P_{AE}}, \tag{17}$$

where, P_A is the static component of the active force and can be calculated by substituting $\theta = 0$ in Eq. (13). Moreover, ΔP_{AE} is the difference between dynamic and static active earth pressure as shown in the following equation:

$$\Delta P_{AE} = P_{AE} - P_A . \tag{18}$$

According to the M-O theory, the total passive earth force under seismic load can be obtained using the following formula [26]:

$$P_{PE} = \frac{1}{2}\gamma H^2 \left(1 - K_V\right) K_{PE}, \tag{19}$$

$$K_{PE} = \frac{\sin^2\left(\alpha - \varnothing - \theta\right)}{\cos(\theta)\sin^2(\alpha)\sin(\alpha + \delta - \theta)\left[1 - \sqrt{\frac{\sin(\delta + \varnothing)\sin(\varnothing + \beta - \theta)}{\sin(\alpha + \delta - \theta)\sin(\alpha + \beta)}}\right]^2} \tag{20}$$

6 Optimization of retaining structures

The aim of the optimum design of retaining structures is to define the design variables related to the least possible value of the objective function, which may be considered as total cost or total weight of the structure while satisfying some stability and strength constraints. In the current study, the total cost of the structure subjected to static and dynamic loads are considered as the objective function based on the following equation:

$$f_{cost} = C_s W_{st} + C_c V_c \,, \tag{21}$$

where, W_{st} is the weight of the steel bars, C_s and C_c are unit cost of steel and concrete, respectively and W_c is the volume of the concrete.

The eight continuous design variables considered here, include five variables related to the geometry of the structure and three more variables representing the steel reinforcement of different parts of the structures depicted graphically in Fig. 1. In this figure, X_1 is width of the heel, X_2 is stem thickness at the top, X_3 is stem thickness at the bottom, X_4 is width of the toe and X_5 is thickness of the base slab, R_1 is the vertical steel reinforcement in the stem, R_2 is the horizontal steel reinforcement in the toe and R_3 is the horizontal steel reinforcement in the heel. Finally, the design constraints implemented by the American Concrete Institute (ACI 318-05) design code [30], considered in the optimization of the retaining structures are summarized in Table 1.

Table 1 Design constraints

Failure mode	Constraints	Considerations
Sliding stability	$FS_S \le (\Sigma F_R / \Sigma F_d)$	
Overturning stability	$FS_O \le (\Sigma M_R / \Sigma M_O)$	
Bearing capacity	$FS_b \le (q_{ult} / q_{max})$	$q_{max,min} = \frac{\Sigma V}{B}\left(1 \pm \frac{6e}{B}\right)$
Eccentricity failure	$e \le (B/6)$	$e = \frac{B}{2} - \frac{\Sigma M_R - \Sigma M_O}{\Sigma V}$
Toe shear	$V_{ut} \le V_{nt}$	$V_n \le \frac{1}{6} 0.75 \sqrt{f_c'} bd$
Toe moment	$M_{ut} \le M_{nt}$	$M_n \le 0.9 A_s f_y \left(d - \frac{a}{2}\right)$, $a = \frac{A_s f_y}{0.85 f_c b}$
Heel shear	$V_{uh} \le V_{nh}$	
Heel moment	$M_{uh} \le M_{nh}$	
Shear at bottom of stem	$V_{us} \le V_{ns}$	
Moment at bottom of stem	$M_{us} \le M_{ns}$	
Limitation of flexural reinforcement	$\rho_{min} \le \rho \le \rho_{max}$	$\rho = \frac{A_s}{bd}$, $\rho_{min} = \frac{1.4}{fy}$, $\rho_{max} = \left(\frac{0.85^2 f_c}{f_y}\right)\left(\frac{600}{600 + f_y}\right)$

In Table 1, FS_S = required factor of safety against sliding; FS_O = required factor of safety against overturning; FS_b = required factor of safety against bearing capacity; $\sum F_R$ = sum of the horizontal resisting forces; $\sum F_d$ = sum of the horizontal driving forces; $\sum M_R$ is sum of the moments of forces that tends to resist overturning about the toe and $\sum M_O$ is sum of the moments of forces that tends to overturn the structure about the toe. $\sum V$ is sum of the vertical forces due to the weight of wall, the soil above the base, and surcharge load. e is the eccentricity, V_{ut}, V_{uh} and V_{us} = ultimate shearing force of toe, heel and stem; V_{nt}, V_{nh} and V_{ns} = nominal shear strength of concrete [30]; M_{ut}, M_{uh} and M_{us} = ultimate bending moment of toe, heel and stem; M_{nt}, M_{nh} and M_{ns} = nominal flexural strength of concrete [30].

7 Comparative analysis of the PSSCA

In this study, the performance of PSSCA is evaluated on a set of unimodal, multimodal and fixed-dimension multimodal benchmark functions from literature [31, 32] against a good combination of some well-known state of the art algorithms. All of these functions are minimization problems, which are useful for evaluating the search efficiency and convergence rate of optimization algorithms. The mathematical formulation and characteristics of these test functions are available in Table 2. The proposed algorithm is coded in MATLAB R2020b programming software.

In this paper, the performance of the proposed PSCA is compared with other well-established optimization algorithms such as the Sine-Cosine Algorithm (SCA) [25], Gravitational Search Algorithm (GSA) [33], Tunicate Swarm Algorithm (TSA) [34] and Grey Wolf Optimizer (GWO) [35]. These algorithms have proved their effectiveness and robustness compared with other methods like Particle Swarm Optimization [25, 33–35].

It should be noted that the performance and convergence of these metaheuristic methods are completely dependent on the internal parameters of the algorithms. PSSCA needs only two main parameters, N (number of objects) and t_{max} (maximum number of iteration). It is found through experiments that lower value of N results in premature convergence and higher value improves exploration but increases elapsed time significantly. The proper value of N is equal to 30 and the maximum number of iteration is considered as 1000. In Table 3, the key parameters of the selected methods are presented. These values have been determined using the reference-based parameter identification process according to the previously published research papers.

Table 2 Description of unimodal benchmark functions

Function	Range	$f_{\min}$
$F_1(X)=\sum_{i=1}^{n} x_i^2$	$[-100,100]^{30}$	0
$F_2(X)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^{30}$	0
$F_3(X)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-100,100]^{30}$	0
$F_4(X)=\max_i\left\{\lvert x_i\rvert, 1\le i\le n\right\}$	$[-100,100]^{30}$	0
$F_5(X)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	$[-30,30]^{30}$	0
$F_6(X)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	$[-100,100]^{30}$	0
$F_7(X)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	$[-500,500]^{30}$	$428.98\times n$
$F_8(X)=\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	$[-5.12,5.12]^{30}$	0
$F_9(X)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	$[-32,32]^{30}$	0
$F_{10}(X)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^{30}$	0
$F_{11}(X)=\frac{\pi}{n}\left\{10\sin\left(\pi y_1\right)+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]+\left(y_n-1\right)^2\right\}+\sum_{i=1}^{n}u\left(x_i,10,100,4\right)$	$[-50,50]^{30}$	0
$F_{12}(X)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\left(x_i-a_{ij}\right)^6}\right)^{-1}$	$[-65.53,65.53]^{30}$	1
$F_{13}(X)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$	$[-5,5]^{4}$	0.00030
$F_{14}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$[-5,5]^{2}$	–1.0316
$F_{15}(X)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)\right)^2$	$[1,3]^{3}$	–3.86
$F_{16}(X)=-\sum_{i=1}^{7}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	$[0,10]^{4}$	–10.4028

Because of stochastic nature of the metaheuristics methods, the results of single run might be unreliable and the algorithms may obtain better or worse solutions than the previously reached one. Therefore, statistical analysis should be applied to have a fair comparison and effectiveness evaluation of the algorithms. Regarding this issue, for the selected algorithms, 30 independent runs are performed and statistical results are collected and reported in Table 4. (Fig. 2–17).

Results of Table 4 show the Best (Minimum), Worst (Maximum), Mean (Average), Median, and Standard Deviation (Std) of the solutions obtained from experiments using the selected optimization algorithms. The best results between the five methods are shown in bold face.

Unimodal test functions can be considered to investigate the exploitation capability of an optimization algorithm [35, 36]. In this study, to evaluate the ability of PSSCA to exploit the promising regions, 6 unimodal benchmark functions (F_1 to F_6) are solved and results are compared with four selected optimization methods in Table 4. The results of this table show that, for all unimodal functions except F_6, PSSCA could provide better solution. In addition, PSSCA can reach the global minimum for F_1–F_4. It means that the new algorithm has a large potential search space compared with the other optimization algorithms.

Multimodal functions with several local optima can be used to evaluate the capability of an algorithm to explore the search space [35, 36]. In this study, 10 multimodal functions (F_7 to F_{16}) are minimized based on the presented procedure. According to the results of Table 4, it can be observed that the Best and Mean values reached by PSSCA for most of the functions (except F_{11}) are significantly better than the other methods. However, for F_{11}, the Mean value obtained by PSSCA are smaller than the robust GSA and results are much better than those obtained by SCA, TSA and GWO. The consistent performance of the new method for suite of multimodal benchmark functions verifies its

Table 3 parameter setting of the selected algorithms

Algorithm	Parameter	Specifications
PSSCA	Number of objects	30
	Maximum iteration	1000
GSA	Search agents	50
	Gravitational constant	100
	Alpha coefficient	20
	Number of generations	1000
GWO	Search agents	80
	Control parameter ($\vec{a}$)	[2,0]
	Number of generations	1000
SCA	Search agents	80
	Number of elites	2
	Number of generations	1000
TSA	Search agents	80
	Parameter P_{min}	1
	Parameter P_{max}	4
	Number of generations	1000
PSO	Search agents	50
	C_1 and C_2	2
	w_{max}	0.9
	w_{min}	0.4
	Number of generations	1000

Table 4 Comparison of different methods in solving test functions

Function	Statistics	PSSCA	SCA	GSA	TSA	GWO
F_1	Best	**0.00**	1.5523e-07	1.0013e-17	5.1458e-61	2.4915e-61
	Worst	**0.00**	0.0043	3.1868e-17	1.1586e-54	3.8647e-58
	Mean	**0.00**	2.3458e-04	2.1148e-17	8.3155e-56	4.9162e-59
	Median	**0.00**	1.9737e-05	2.0077e-17	7.1012e-58	1.0534e-59
	Std.	**0.00**	7.9295e-04	5.8150e-18	2.4905e-55	1.0230e-58
F_2	Best	**0.00**	1.5005e-09	1.5282e-08	1.1196e-35	8.3612e-36
	Worst	**0.00**	9.8446e-06	3.3313e-08	3.2814e-32	5.3488e-34
	Mean	**0.00**	1.6882e-06	2.3935e-08	2.1532e-33	8.3658e-35
	Median	**0.00**	5.4000e-07	2.3469e-08	3.1044e-34	5.9294e-35
	Std.	**0.00**	2.4046e-06	4.0025e-09	6.0237e-33	9.8594e-35
F_3	Best	**0.00**	70.8285	102.9550	2.5684e-32	1.2533e-19
	Worst	**0.00**	2.6762e+03	468.6160	2.4492e-17	3.5572e-13
	Mean	**0.00**	789.1620	245.4694	8.1741e-19	1.5096e-14
	Median	**0.00**	619.4506	221.1150	1.8696e-24	2.0740e-17
	Std.	**0.00**	746.2287	100.1024	4.4714e-18	6.5547e-14
F_4	Best	**0.00**	1.2610	2.2498e-09	3.2458e-08	9.8174e-16
	Worst	**0.00**	35.6743	5.0857e-09	6.3429e-05	2.4431e-13
	Mean	**0.00**	9.3080	3.3030e-09	1.0102e-05	1.9487e-14
	Median	**0.00**	6.9806	3.2020e-09	2.0270e-06	6.3817e-15
	Std.	**0.00**	8.0720	7.4424e-10	1.6927e-05	4.4955e-14

Continuation of **Table 4**

Function	Statistics	PSSCA	SCA	GSA	TSA	GWO
F_5	Best	**3.5924e-04**	27.3230	25.7459	25.6273	25.2273
	Worst	**3.5924e-04**	49.5110	220.9110	29.5430	28.7294
	Mean	**3.5924e-04**	29.9106	42.2647	28.4422	26.9256
	Median	**3.5924e-04**	29.0097	26.1443	28.8115	27.1173
	Std.	**1.6541e-19**	4.1508	45.4674	0.7616	0.8418
F_6	Best	1.9836e-07	3.4070	**9.711e-18**	2.0585	0.2466
	Worst	0.0220	4.4435	**8.645e-16**	4.7791	1.2619
	Mean	0.0021	4.0360	**3.097e-17**	3.6724	0.6376
	Median	1.9836e-07	4.0572	**2.953e-17**	3.5615	0.7452
	Std.	0.0056	0.2954	**6.165e-18**	0.6918	0.3353
F_7	Best	**-1.2050e+04**	-5.2993e+03	-3.6279e+03	-7.8992e+03	-8.8178e+03
	Worst	**-1.1096e+04**	-3.5321e+03	-2.0033e+03	-5.2761e+03	-4.9742e+03
	Mean	**-1.2005e+04**	-4.0769e+03	-2.7826e+03	-6.6126e+03	-6.2524e+03
	Median	**-1.2050e+04**	-3.9720e+03	-2.7464e+03	-6.6131e+03	-6.2270e+03
	Std.	**186.4737**	336.8249	365.4671	599.2609	852.4634
F_8	Best	**0.00**	1.0560e-06	8.9546	77.7761	0.00
	Worst	**0.00**	51.4451	21.8891	254.9883	10.0548
	Mean	**0.00**	5.9694	15.6209	151.4539	0.8853
	Median	**0.00**	9.3391e-04	15.9193	149.6596	0.00
	Std.	**0.00**	12.2476	3.1043	35.8717	2.4438
F_9	Best	**8.8818e-16**	1.5579e-05	2.5288e-09	1.5099e-14	1.1546e-14
	Worst	**8.8818e-16**	20.2198	4.4823e-09	4.3125	2.2204e-14
	Mean	**8.8818e-16**	14.3622	3.4912e-09	2.4095	1.5928e-14
	Median	**8.8818e-16**	20.1275	3.4766e-09	2.9381	1.5099e-14
	Std.	**0.00**	8.9778	5.1530e-10	1.3920	2.5861e-15
F_{10}	Best	**0.00**	4.8381e-07	1.6952	0.00	0.00
	Worst	**0.00**	0.7703	10.6642	0.0159	0.0140
	Mean	**0.00**	0.1368	4.2510	0.0077	0.0014
	Median	**0.00**	0.0032	3.5667	0.0082	0.00
	Std.	**0.00**	0.2218	2.0234	0.0057	0.0041
F_{11}	Best	3.9317e-08	0.2631	**8.2033e-20**	0.2738	0.0121
	Worst	**1.5374e-04**	5.6300	0.1037	13.8088	0.0920
	Mean	**7.0132e-06**	0.9568	0.0198	6.3735	0.0364
	Median	4.0116e-07	0.4964	**1.3512e-19**	6.7411	0.0329
	Std.	**2.7947e-05**	1.1497	0.0400	3.4586	0.0201
F_{12}	Best	**0.9980**	**0.9980**	**0.9980**	**0.9980**	**0.9980**
	Worst	**0.9980**	2.9821	8.0858	12.6705	12.6705
	Mean	**0.9980**	1.1964	3.6212	7.6657	4.1312
	Median	**0.9980**	0.9980	3.0452	10.7632	2.9821
	Std.	**1.4772e-11**	0.6054	2.1942	4.8845	4.1443
F_{13}	Best	**3.1381e-04**	3.4063e-04	0.0012	3.751e-04	3.1749e-04
	Worst	**3.9684e-04**	0.0014	0.0118	0.0566	0.0204
	Mean	**3.3641e-04**	8.5975e-04	0.0025	0.0043	0.0044
	Median	**3.2323e-04**	7.3095e-04	0.0021	4.5390e-04	3.0754e-04
	Std.	**2.4589e-05**	3.8089e-04	0.0019	0.0116	0.0081
F_{14}	Best	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
	Worst	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
	Mean	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
	Median	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
	Std.	1.8597e-06	1.0395e-05	5.6082e-05	0.0058	**4.7385e-09**
F_{15}	Best	**-3.8628**	-3.8625	**-3.8628**	**-3.8628**	**-3.8628**
	Worst	**-3.8628**	-3.8539	**-3.8628**	-3.8549	-3.8549
	Mean	**-3.8628**	-3.8560	**-3.8628**	-3.8625	-3.8620
	Median	**-3.8628**	-3.8548	**-3.8628**	-3.8628	-3.8628
	Std.	**1.3625e-16**	0.0029	2.4795e-05	0.0014	0.0022
F_{16}	Best	**-10.4028**	-9.0513	-10.4009	-10.3812	-10.4029
	Worst	**-10.4028**	-0.9074	-10.4029	-2.7427	-5.0877
	Mean	**-10.4028**	-5.4154	-10.4029	-7.8325	-10.2253
	Median	**-10.4028**	-5.0380	-10.4028	-10.2554	-10.4025
	Std.	**5.4202e-15**	1.7315	4.6649e-06	3.1843	0.9703

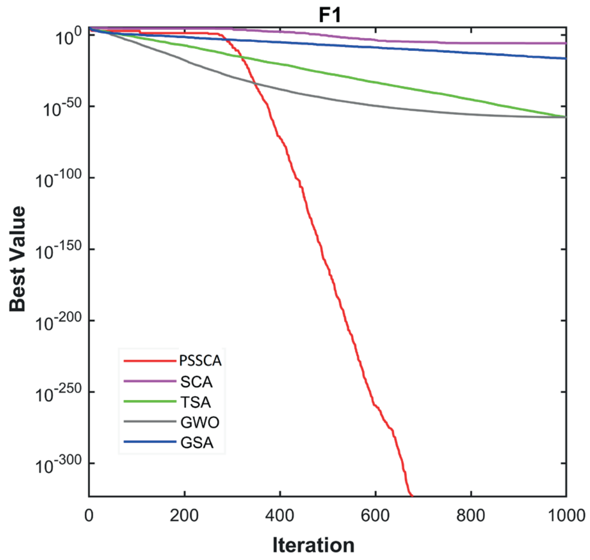

Fig. 2 Convergence curves of algorithms for F_1

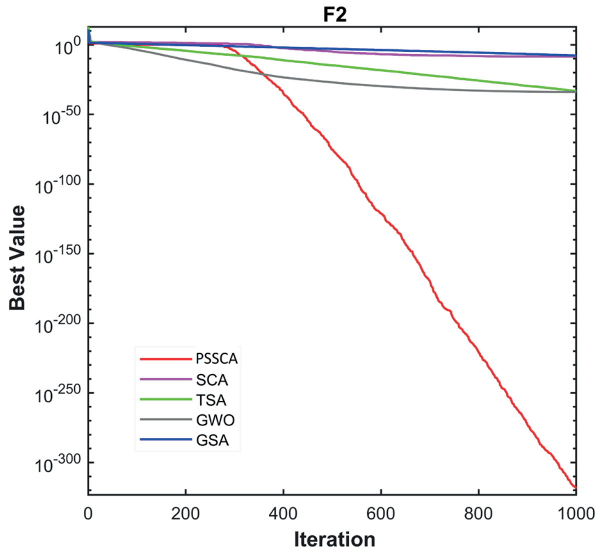

Fig. 3 Convergence curves of algorithms for F_2

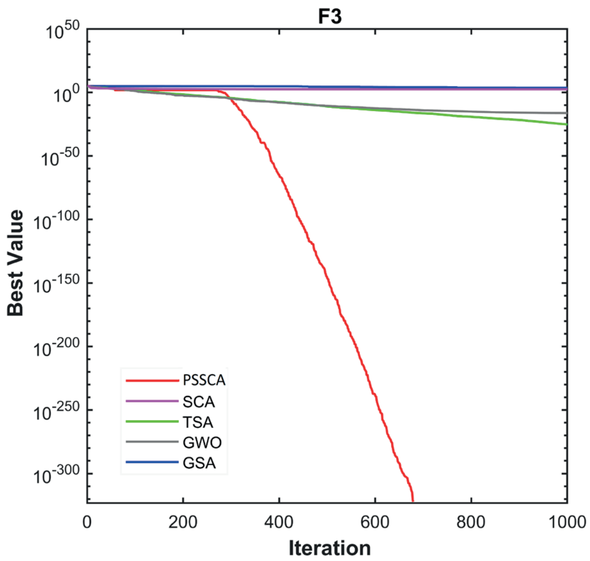

Fig. 4 Convergence curves of algorithms for F_3

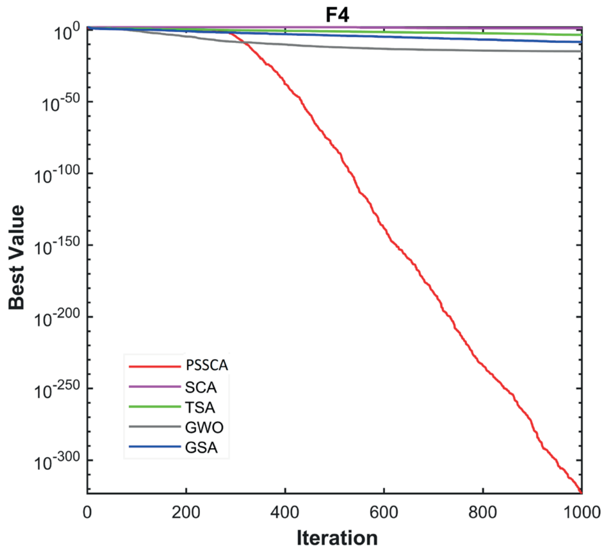

Fig. 5 Convergence curves of algorithms for F_4

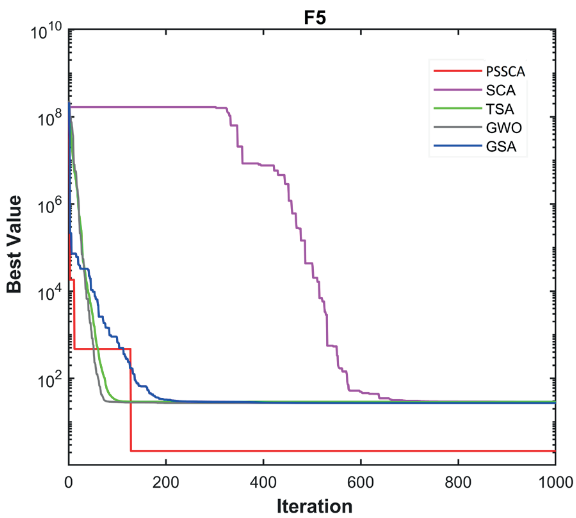

Fig. 6 Convergence curves of algorithms for F_5

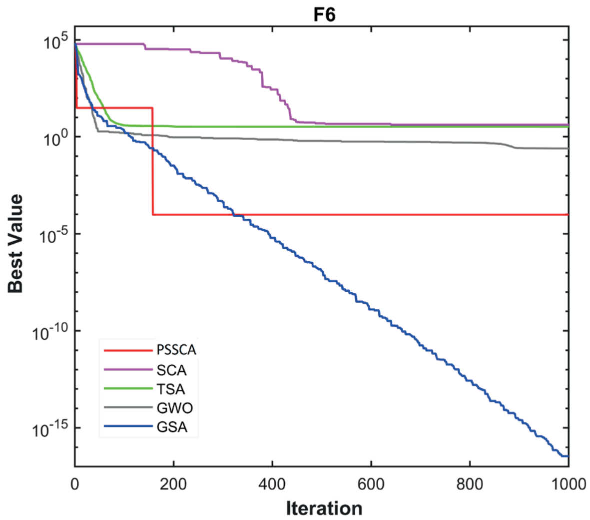

Fig. 7 Convergence curves of algorithms for F_6

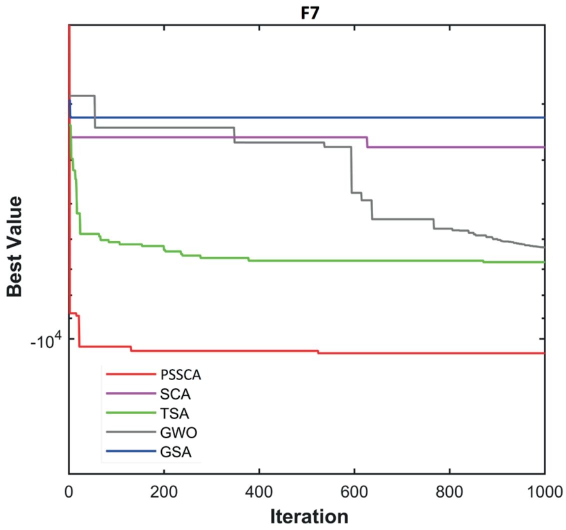

Fig. 8 Convergence curves of algorithms for F_7

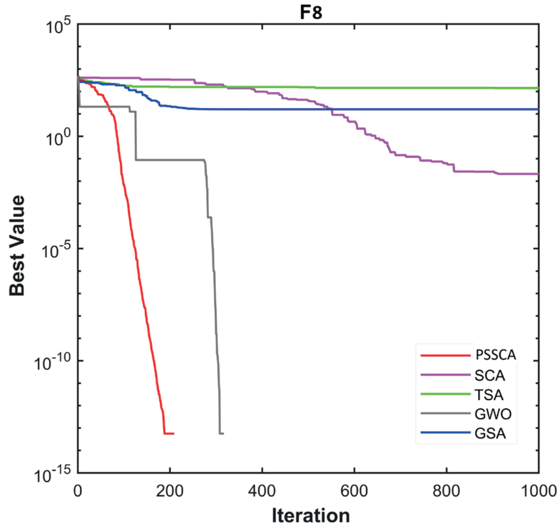

Fig. 9 Convergence curves of algorithms for F_8

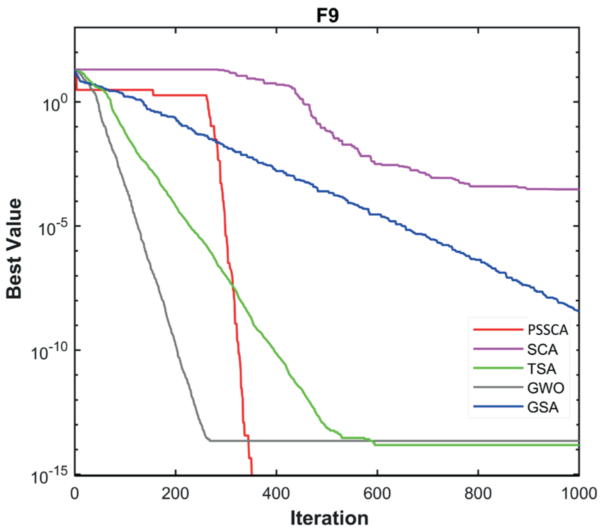

Fig. 10 Convergence curves of algorithms for F_9

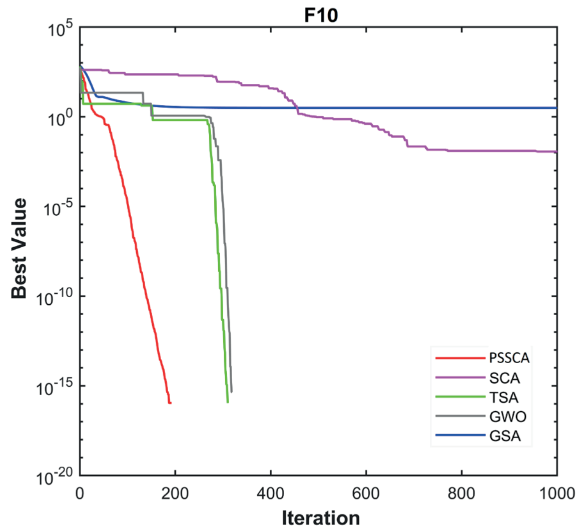

Fig. 11 Convergence curves of algorithms for F_{10}

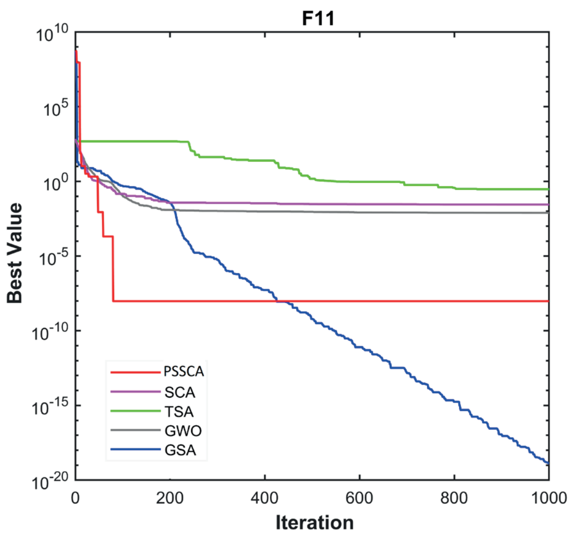

Fig. 12 Convergence curves of algorithms for F_{11}

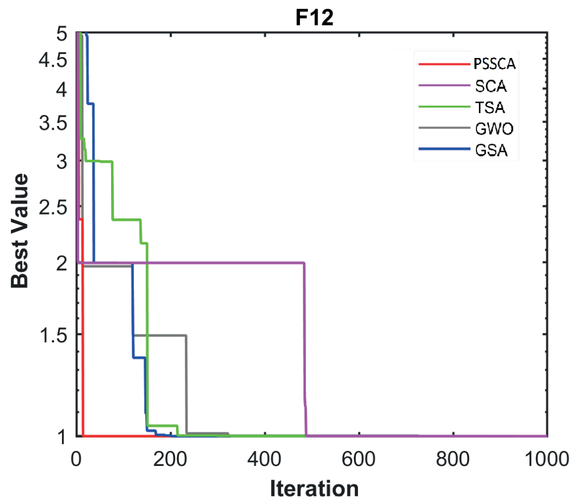

Fig. 13 Convergence curves of algorithms for F_{12}

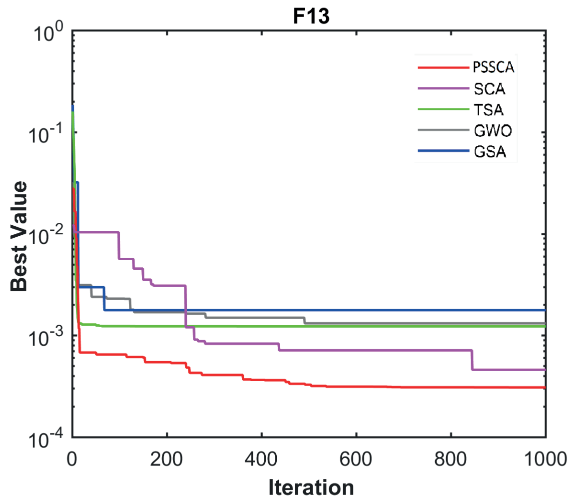

Fig. 14 Convergence curves of algorithms for F_{13}

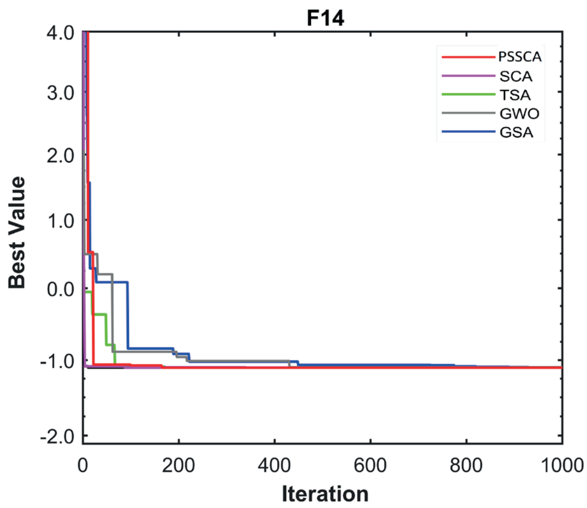

Fig. 15 Convergence curves of algorithms for F_{14}

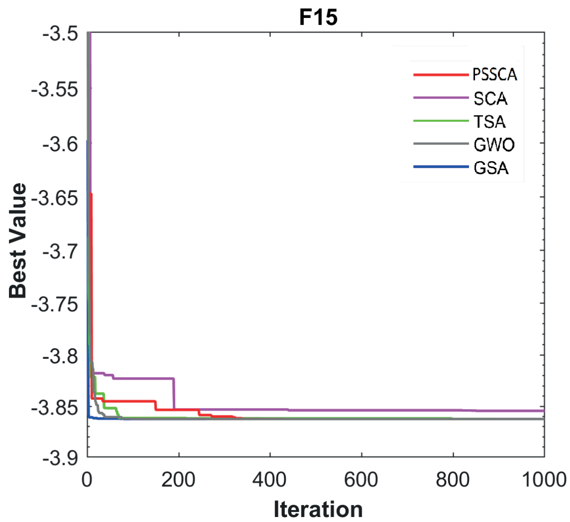

Fig. 16 Convergence curves of algorithms for F_{15}

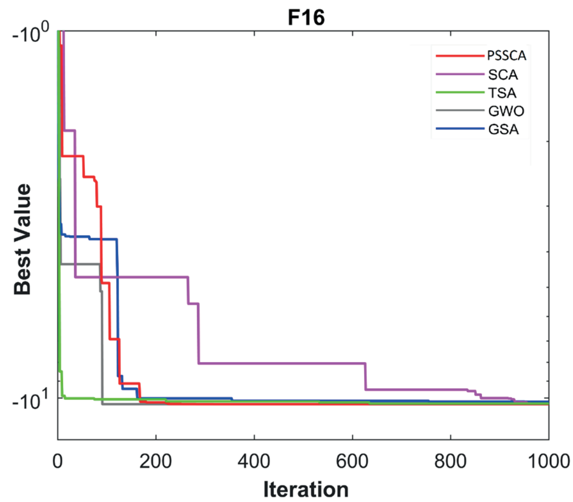

Fig. 17 Convergence curves of algorithms for F_{16}

superior capabilities of exploration. From the standard deviation point of view, which indicate the stability of the algorithm, the results show that PSSCA is a more stable method when compared with the other techniques. In addition, the convergence progress curves of algorithms for benchmark functions are compared in Fig. 2–17. From the above analysis, it can be concluded that PSSCA either outperforms the other algorithms or performs almost equivalently.

In order to determine the statistical significance of the comparative results of two or more algorithms, a non-parametric pairwise statistical analysis should be conducted. As recommended by Derrac et al. [37] to assess meaningful comparison between the proposed and alternative methods, the nonparametric Wilcoxon's rank sum test is performed between the results. In this regard, utilizing the best results obtained from 30 runs of each method, a pairwise comparison is conducted.

Wilcoxon's rank sum test returns p-value, sum of positive ranks (R+) and the sum of negative ranks (R−). Table 5 presents the results of Wilcoxon's rank sum test of PSSCA when compared with other methods. The p-value indicates the minimum of significance level for detecting differences. In this study, $\alpha = 0.05$ is considered as the level of significance. If the p-value of the given algorithm is bigger than 0.05, then there is no significant difference between the two compared methods. Such a result indicated with "N.A" in the winner rows of Table 5. On the other hand, if the p-value is less than α, it definitively means that, in each pair-wise comparison, the better result obtained by the best algorithm is statistically significant and it was not gained by chance. In such cases, if the R+ is bigger than R–, indicates PSSCA has a superior performance than the alternative method otherwise PSSCA has inferior performance and alternative algorithm shown better performance [38].

Table 5 results of Wilcoxon's rank sum test

Function	Wilcoxon test Parameters	PSSCA vs GSA	PSSCA vs SCA	PSSCA vs TSA	PSSCA vs GWO
F_1	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_2	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_3	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_4	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_5	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_6	p- value	7.4523E-7	1.7344E-06	1.7344E-06	2.353E-06
	R+	0	465	465	462
	R-	465	0	0	3
	Winner	GSA	PSSCA	PSSCA	PSSCA
F_7	p- value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_8	p- value	1.7344E-06	1.7344E-06	1.7344E-06	0.02
	R+	465	465	465	21
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_9	p- value	1.73E-06	1.73E-06	1.73E-06	1.73E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_{10}	p- value	1.73E-06	1.73E-06	1.473E-03	0.068
	R+	465	465	91	10
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	N.A
F_{11}	p- value	0.041	1.73E-06	1.73E-06	1.73E-06
	R+	140	465	465	465
	R-	325	0	0	0
	Winner	GSA	PSSCA	PSSCA	PSSCA
F_{12}	p- value	1.73E-06	2.56E-06	4.81E-06	3.22E-04
	R+	465	435	403	152
	R-	0	0	3	1
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_{13}	p- value	1.73E-06	2.35E-06	0.006	0.393
	R+	465	462	366	274
	R-	0	3	99	191
	Winner	PSSCA	PSSCA	PSSCA	N.A
F_{14}	p- value	0.059	0.371	0.132	1.59E-06
	R+	304	276	50	0
	R-	161	189	415	465
	Winner	N.A	N.A	N.A	GWO

Continuation of **Table 5**

Function	Wilcoxon test Parameters	PSSCA vs GSA	PSSCA vs SCA	PSSCA vs TSA	PSSCA vs GWO
F_{15}	p- value	8.38E-08	1.73E-06	1.73E-06	1.73E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
F_{16}	p- value	1.44E-07	1.73E-06	1.73E-06	2.13E-06
	R+	465	465	465	463
	R-	0	0	0	2
	Winner	PSSCA	PSSCA	PSSCA	PSSCA
Total	Superior /Inferior/N.A	13/2/1	15/0/1	15/0/1	13/1/2

8 Model application

In this section, two numerical examples of retaining structures are considered for investigating the efficiency of the proposed algorithm.

These experiments are solved by considering three different combinations of K_h and K_V ; $K_h = 0$, $K_V = 0$, $K_h = 0.2$, $K_V = 0$ and $K_h = 0.2$ and $K_V = 0.2$. Table 6 presents input parameters for these experiments.

The first example is originally presented by Saribas and Erbatur [39] and has been solved using classical nonlinear programming (NLP). Camp and Akin [40] applied big bang-big crunch (BB-BC) optimization method and Gandomi et al. [17] developed interior search algorithm (ISA) for the solution. However, all these research solved the problem under static loading condition which is equivalent with first case in the current study ($K_h = 0$, $K_V = 0$). The problem is solved using the proposed PSSCA as well as SCA, GSA, TSA and GWO for different combination of horizontal and vertical acceleration coefficient. The algorithm is run 30 times and the best results are presented in Table 7.

As the results of Table 7 show, the best value of the objective function obtained by PSSCA for the first case (static loading) is 69.035 $/m which is approximately 20% cheaper than the design presented by Saribas and Erbatur [39], 2% lower than BB-BS and 5.8% cheaper than the ISA method. In addition, the best cost evaluated by PSSCA are slightly lower than those obtained by SCA, TSA and GWO for all loading cases. The results of GSA are comparable with PSSCA, while the required computation time of GSA is more than the new hybrid algorithm.

Moreover, the results show that considering seismic condition ($K_h = 0.2$, $K_V = 0$) will increase the construction cost around 19% and by increasing K_V to 0.2, the best cost will decrease slightly as it was predictable from Eq. (12).

Similarly, the second example has been studied previously by Saribas and Erbatur [39] using classical nonlinear programming (NLP) and Gandomi et al. [17] using interior

Table 6 Input parameters for numerical Example 1 and 2

parameter	Unit	Symbol	Value for Example 1	Value for Example 2
Height of stem	m	H	3.0	4.5
Internal friction angle of retained soil	degree	φ	36	36
Internal friction angle of base soil	degree	φ'	0.0	34
Unit weight of retained soil	kN/m^3	γ_s	17.5	17.5
Unit weight of base soil	kN/m^3	γ_s'	18.5	18.5
Unit weight of concrete	kN/m^3	γ_c	23.5	23.5
Unit weight of steel	kN/m^3	γ_{steel}	78.5	78.5
Cohesion of base soil	kPa	c	125	0.0
Depth of soil in front of wall	m	D	0.5	0.75
Surcharge load	kPa	q	20	30
Backfill slope	degree	β	10	15
Concrete cover	cm	d_c	7.0	7.0
Yield strength of reinforcing steel	MPa	f_y	400	400
Compressive strength of concrete	MPa	f_c	21	21
Shrinkage and temporary reinforcement percent	-	ρ_{st}	0.002	0.002
Design load factor	-	LF	1.7	1.7
Factor of safety for overturning stability	-	FS_O	1.5	1.5
Factor of safety against sliding	-	FS_S	1.5	1.5
Factor of safety for bearing capacity	-	FS_B	3.0	3.0
Cost of steel	$/kg	C_S	0.4	0.4
Cost of concrete	$/m^3	C_C	40	40

Table 7 Optimization result for design Example 1

Design variable	Unit	Optimum values $K_h = 0$, $K_V = 0$	Optimum values $K_h = 0.2$, $K_V = 0$	Optimum values $K_h = 0.2$, $K_V = 0.2$
X_1	m	0.6498	0.8969	0.8659
X_2	m	0.2	0.2	0.2
X_3	m	0.28	0.3	0.295
X_4	m	0.6843	0.7778	0.6996
X_5	m	0.2727	0.2918	0.3105
R_1	cm²/m	12.5	13.3	12
R_2	cm²/m	16.4	21.6	19.5
R_3	cm²/m	16.6	21.6	20
PSSCA	$/m	69.035	82.192	78.67
SCA	$/m	70.43	84.56	78.86
GSA	$/m	69.088	82.187	78.62
TSA	$/m	69.142	82.231	79.12
GWO	$/m	72.81	85.21	81.32
NLP [39]	$/m	82.474	-	-
BB-BC [40]	$/m	70.38	-	-
ISA [17]	$/m	73.05	-	-

Table 8 Optimization result for design Example 2

Design variable	Unit	Optimum values $K_h = 0$, $K_V = 0$	Optimum values $K_h = 0.2$, $K_V = 0$	Optimum values $K_h = 0.2$, $K_V = 0.2$
X_1	m	1.6757	1.8734	1.8
X_2	m	0.2274	0.2787	0.2619
X_3	m	0.4167	0.4454	0.424
X_4	m	0.8862	1.1667	1.1667
X_5	m	0.4504	0.5184	0.5089
R_1	cm²/m	21	22	20.7
R_2	cm²/m	39	42	40
R_3	cm²/m	38	42	39
PSSCA	$/m	185.2	223.95	210.58
SCA	$/m	187.24	226.65	213.13
GSA	$/m	185.2	224.11	210.86
TSA	$/m	186.8	224.92	212.32
GWO	$/m	188.95	227.86	212.95
NLP [39]	$/m	189.55	_	_
ISA [17]	$/m	190.06	_	_

search algorithm (ISA). This case is solved using the proposed PSSCA as well as mentioned algorithms in Section 7 for different values of K_h and K_V. The algorithms have been run 30 times and the best results presented in Table 8. Based on the results, the new method could provide better solution compared with the other methods, which indicate the consistent performance of the PSSCA. Moreover, the obtained results reveal that by increasing K_h to 0.2, the best cost will be increased 21%. While, by increasing K_V, the inverse trend of this state becomes apparent and the best price slightly decreased. Therefore, in the prevalent conditions of the seismic optimization of the retaining structure, ignoring the K_V is acceptable.

9 Conclusions

This study develops a novel hybrid algorithm, namely PSSCA, by integrating PSO and SCA techniques. In the proposed PSSCA algorithm, during the search process the candidate solutions interact with each other and improve their positions based on the best position obtained so far as the reference point. In summary, the main features of PSSCA are as follows: it has just two internal parameters; it is easy to code; and it is easy to apply. The performance of the proposed algorithm is benchmarked using a set of six unimodal and ten multi-modal test functions and the results were compared with four well-known and recently developed algorithms including GSA, SCA, TSA and GWO. According to the results and finding, it was observed and may be concluded that PSSCA is capable of finding the global solution for most of the unimodal and multi-modal benchmark functions and it outperforms the other algorithms in a statistically significant manner. Finally, the performance of the new algorithm for low-cost design of retaining structures under static and seismic loading conditions is investigated through two numerical examples and obtained results were compared with other methods. The numerical experiments reveal that the newly proposed algorithm for optimum design of retaining structures is quite robust and efficient when compared with the other techniques.

References

[1] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. "Optimization by Simulated Annealing", Science, 220(4598), pp. 671–680, 1983. https://doi.org/10.1126/science.220.4598.671

[2] de Werra, D., Hertz, A. "Tabu search techniques", Operations-Research-Spektrum, 11(3), pp. 131–141, 1989. https://doi.org/10.1007/BF01720782

[3] Lourenço, H. R., Martin, O. C., Stützle, T. "Iterated Local Search: Framework and Applications", In: Gendreau, M., Potvin, J.-Y. (eds.) Handbook of Metaheuristics, Springer, Cham, Switzerland, 2019, pp. 129–168. https://doi.org/10.1007/978-3-319-91086-4_5

[4] Doğan, B., Ölmez, T. "A new metaheuristic for numerical function optimization: Vortex Search algorithm", Information Sciences, 293, pp. 125–145, 2015.
https://doi.org/10.1016/j.ins.2014.08.053

[5] Holland, J. H. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", MIT Press, Cambridge, MA, USA, 1992.
https://doi.org/10.7551/mitpress/1090.001.0001

[6] Dorigo, M., Di Caro, G. "Ant colony optimization: a new meta-heuristic", In: Proceedings of the 1999 Congress on Evolutionary Computation - CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1470–1477.
https://doi.org/10.1109/CEC.1999.782657

[7] Kennedy, J., Eberhart, R. "Particle swarm optimization", In: Proceedings of ICNN'95 International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948.
https://doi.org/10.1109/ICNN.1995.488968

[8] Geem, Z. W., Kim, J. H., Loganathan, G. V. "A New Heuristic Optimization Algorithm: Harmony Search", Simulation, 76(2), pp. 60–68, 2001.
https://doi.org/10.1177/003754970107600201

[9] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. "Harris hawks optimization: Algorithm and applications", Future Generation Computer Systems, 97, pp. 849–872, 2019.
https://doi.org/10.1016/j.future.2019.02.028

[10] Wolpert, D. H., Macready, W. G. "No free lunch theorems for optimization", IEEE Transactions on Evolutionary Computation, 1(1), pp. 67–82, 1997.
https://doi.org/10.1109/4235.585893

[11] Kaveh, A., Ilchi Ghazaan, M., Asadi, A. "An Improved Water Strider Algorithm for Optimal Design of Skeletal Structures", Periodica Polytechnica Civil Engineering, 64(4), pp. 1284–1305, 2020.
https://doi.org/10.3311/PPci.16872

[12] Cheng, Y.-M. "Modified Harmony Methods for Slope Stability Problems", Springer, Springer, Berlin, Heidelberg, Germany, 2009.
https://doi.org/10.1007/978-3-642-00185-7_9

[13] Dhiman, G., Garg, M., Nagar, A., Kumar, V., Dehghani, M. "A novel algorithm for global optimization: Rat Swarm Optimizer", Journal of Ambient Intelligence and Humanized Computing, 12, pp. 1–26, 2020.
https://doi.org/10.1007/s12652-020-02580-0

[14] Khajehzadeh, M., Taha, M. R., Eslami, M. "A New Hybrid Firefly Algorithm for Foundation Optimization", National Academy Science Letters, 36(3), pp. 279–288, 2013.
https://doi.org/10.1007/s40009-013-0129-z

[15] Khajehzadeh, M., Taha, M. R., El-Shafie, A. "Reliability analysis of earth slopes using hybrid chaotic particle swarm optimization", Journal of Central South University of Technology, 18(5), pp. 1626–1637, 2011.
https://doi.org/10.1007/s11771-011-0882-4

[16] Kalemci, E. N., İkizler, S. B., Dede, T., Angın, Z. "Design of reinforced concrete cantilever retaining wall using Grey wolf optimization algorithm", Structures, 23, pp. 245–253, 2020.
https://doi.org/10.1016/j.istruc.2019.09.013

[17] Gandomi, A. H., Kashani, A. R., Zeighami, F. "Retaining wall optimization using interior search algorithm with different bound constraint handling", International Journal for Numerical and Analytical Methods in Geomechanics, 41(11), pp. 1304–1331, 2017.
https://doi.org/10.1002/nag.2678

[18] Khajehzadeh, M., Taha, M. R., Eslami, M. "Efficient gravitational search algorithm for optimum design of retaining walls", Structural Engineering and Mechanics, 45(1), pp. 111–127, 2013.
https://doi.org/10.12989/sem.2013.45.1.111

[19] Kalateh-Ahani, M., Sarani, A. "Performance-based Optimal Design of Cantilever Retaining Walls", Periodica Polytechnica Civil Engineering, 63(2), pp. 660–673, 2019.
https://doi.org/10.3311/PPci.13201

[20] Khajehzadeh, M., Taha, M. R., Eslami, M. "Multi-objective optimisation of retaining walls using hybrid adaptive gravitational search algorithm", Civil Engineering and Environmental Systems, 31(3), pp. 229–242, 2014.
https://doi.org/10.1080/10286608.2013.853746

[21] Kaveh, A., Hamedani, K. B., Bakhshpoori, T. "Optimal Design of Reinforced Concrete Cantilever Retaining Walls Utilizing Eleven Meta-Heuristic Algorithms: A Comparative Study", Periodica Polytechnica Civil Engineering, 64(1), pp. 156–168, 2020.
https://doi.org/10.3311/PPci.15217

[22] Temur, R. "Optimum design of cantilever retaining walls under seismic loads using a hybrid TLBO algorithm", Geomechanics and Engineering, 24(3), pp. 237–251, 2021.
https://doi.org/10.12989/GAE.2021.24.3.237

[23] Aydogdu, I. "Cost optimization of reinforced concrete cantilever retaining walls under seismic loading using a biogeography-based optimization algorithm with Levy flights", Engineering Optimization, 49(3), pp. 381–400, 2017.
https://doi.org/10.1080/0305215X.2016.1191837

[24] Rahbari, P., Ravichandran, N., Juang, C. H. "Robust Geotechnical Design of a Retaining Wall Subjected to Earthquake Loads", In: Proceedings of the Geotechnical Frontiers 2017, Orlando, Florida, USA, 2017, pp. 149–158.
https://doi.org/10.1061/9780784480458.015

[25] Mirjalili, S. "SCA: a Sine Cosine Algorithm for solving optimization problems", Knowledge-Based Systems, 96, pp. 120–133, 2016.
https://doi.org/10.1016/j.knosys.2015.12.022

[26] Das, B. M., Luo, Z. "Principles of Soil Dynamics", Cengage Learning, Boston, MA, USA, 2016.

[27] Mononobe, N., Matsuo, H. "On determination of earth pressure during earthquake", Proceedings of the World Engineering Congress, Tokyo, Japan, 1929, pp. 177–185.

[28] Okabe, S. "General theory on earth pressure and seismic stability of retaining wall and dam", Journal of Japan Society of Civil Engineering, 10(6), pp. 1277–1323, 1924.

[29] Cakir, T., Livaoglu, R., Dogangün, A. "Effects of different backfill soil types on dynamic behavior of rectangular tank wall considering backfill-wall-fluid interaction", presented at The 14th World Conference on Earthquake Engineering (14WCEE), Beijing, China, Oct. 12–17, 2008.

[30] ACI Committee 318 " ACI 318-05 Building Code Requirements for Structural Concrete", American Concrete Institute, Farmington Hills, MI, USA, 2005.

[31] Digalakis, J. G., Margaritis, K. G. "On benchmarking functions for genetic algorithms", International Journal of Computer Mathematics, 77(4), pp. 481–506, 2001.
https://doi.org/10.1080/00207160108805080

[32] Hussain, K., Salleh, M. N. M., Cheng, S., Naseem, R. "Common benchmark functions for metaheuristic evaluation: A review", JOIV: International Journal on Informatics Visualization, 1(4-2), pp. 218–223, 2017.
https://doi.org/10.30630/joiv.1.4-2.65

[33] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. "GSA: a Gravitational Search Algorithm", Information Sciences, 179(13), pp. 2232–2248, 2009.
https://doi.org/10.1016/j.ins.2009.03.004

[34] Kaur, S., Awasthi, L. K., Sangal, A. L., Dhiman, G. "Tunicate Swarm Algorithm: a new bio-inspired based metaheuristic paradigm for global optimization", Engineering Applications of Artificial Intelligence, 90, Article number: 103541, 2020.
https://doi.org/10.1016/j.engappai.2020.103541

[35] Mirjalili, S., Mirjalili, S. M., Lewis, A. "Grey Wolf Optimizer", Advances in Engineering Software, 69, pp. 46–61, 2014.
https://doi.org/10.1016/j.advengsoft.2013.12.007

[36] Askari, Q., Younas, I., Saeed, M. "Political Optimizer: A novel socio-inspired meta-heuristic for global optimization", Knowledge-Based Systems, 195, Article number: 105709, 2020.
https://doi.org/10.1016/j.knosys.2020.105709

[37] Derrac, J., García, S., Molina, D., Herrera, F. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", Swarm and Evolutionary Computation, 1(1), pp. 3–18, 2011.
https://doi.org/10.1016/j.swevo.2011.02.002

[38] Toz, M. "Chaos-based Vortex Search algorithm for solving inverse kinematics problem of serial robot manipulators with offset wrist", Applied Soft Computing, 89, Article number: 106074, 2020.
https://doi.org/10.1016/j.asoc.2020.106074

[39] Sarıbaş, A., Erbatur, F. "Optimization and Sensitivity of Retaining Structures", Journal of Geotechnical Engineering, 122(8), pp. 649–656, 1996.
https://doi.org/10.1061/(ASCE)0733-9410(1996)122:8(649)

[40] Camp, C. V., Akin, A. "Design of Retaining Walls Using Big Bang–Big Crunch Optimization", Journal of Structural Engineering, 138(3), pp. 438–448, 2012.
https://doi.org/10.1061/(ASCE)ST.1943-541X.0000461