



Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm

Sushmita Sharma¹ · Apu Kumar Saha¹ · Geetanjali Lohar¹

Received: 2 April 2020 / Accepted: 9 January 2021 / Published online: 15 February 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

Abstract

A retaining wall is a structure used to resist the lateral pressure of soil or any backfill material. Cantilever retaining walls provide resistance to overturning and sliding by using backfill weight. In this paper, the weight and cost of the cantilever retaining wall have been minimized using a hybrid metaheuristic optimization technique, namely, h-BOASOS. The algorithm has been developed by the ensemble of two popular metaheuristics, butterfly optimization algorithm (BOA) and symbiosis organism search (SOS) algorithm. BOA's exploratory intensity is coupled with SOS's greater exploitative capacity to find the superior algorithm h-BOASOS. The newly developed algorithm has been tested with a suite of 35 classical benchmark functions, and the results are compared with several state-of-the-art metaheuristic algorithms. The results are evaluated statistically by the Friedman rank test, and convergence curves measure the convergence speed of the algorithm. It is observed in both cases that h-BOASOS is superior to other algorithms. The suggested approach is then used to solve four real-world engineering design problems to examine the problem-solving capacity of the proposed algorithm, and the results are contrasted with a wide range of algorithms. The proposed h-BOASOS is considered to be the winner on each occasion. Finally, the newly suggested algorithm is applied to find the cost and weight of the cantilever retaining wall problems of two different heights, 3.2 m and 6.3 m. The obtained results are compared with the component algorithms and found that the new algorithm works better than the compared algorithms.

Keywords Meta-heuristics · Hybrid algorithm · Butterfly optimization algorithm · Symbiosis organisms search · Cantilever retaining wall

1 Introduction

Soil is unstable in many circumstances due to its integral angle of inclination. There are several procedures to make these slopes stable. One of the most popular methods to stabilize slopes is the use of retaining wall structures. The retaining wall is the structure that holds soil behind it and resists lateral pressure of the soil. Designing a retaining wall depends on three primary criteria: structural strength, geotechnical stability, and the economy. It generally uses a trial

and error approach to satisfy these criteria simultaneously, and these criteria are checked against geotechnical and structural requirements. Thus, these approaches demand several repetitions, and there is no assurance that the final design is the best from an economic point of view.

Optimization algorithms can be exploited to reach a cost-effective design satisfying all the geotechnical and structural requirements simultaneously. It is extremely challenging to find a design that satisfies all the safety criteria, so it is essential to formulate the problem as one of the mathematical non-linear programming techniques. Because of these reasons, these approaches demand several repetitions, which are time-consuming, and there is no assurance that the final design obtained will be safe and best from an economic point of view. Existing optimization structural software used for retaining wall design lacks the ability to find out optimal design because of their deterministic nature, while stochastic methods adopted are not tailored specifically for retaining walls and massive concrete structures. To overcome these

✉ Apu Kumar Saha
apusaha_nita@yahoo.co.in

Sushmita Sharma
snsush01@gmail.com

Geetanjali Lohar
lohargeetanjali@gmail.com

¹ National Institute of Technology, Agartala, India

limitations and find out a cost-effective design, there is a need for an optimization algorithm to solve the retaining wall problem. Numerous studies have investigated the optimization of retaining walls using meta-heuristic algorithms. Hence, it is one of the most compelling research topics in geotechnical and structural optimization.

The retaining wall optimization problem comprises of structural strength and geotechnical stability requirements. The objective function is to minimize both the cost and weight of the retaining wall. To economize the cost, satisfying these situations, there is a need to vary the dimensions of the wall several times, which makes it very tedious and monotonous. It is incredibly challenging to find a design that satisfies all the safety criteria, so it is essential to formulate the problem as one of the mathematical non-linear programming techniques.

Retaining walls are the rigid wall which laterally supports the soil on both the sides at different levels. Retaining walls are further classified into gravity, cantilever, counterfort, and buttress retaining wall. In this study, the cantilever retaining wall problem is considered for optimization. The cantilever retaining wall is constructed of reinforced concrete with a thinner stem. Resistance to overturning and sliding is provided by utilizing the weight of backfill soil.

Due to the high complexity, such as highly non-linearity, multi-modal property, a large number of variables, and constraints of present-day optimization problems, the researchers are inclined to solve these problems by meta-heuristic optimization techniques as the available methods cannot cope with these types of complexities. Due to simplicity, meta-heuristic methods are gaining popularity day by day. The meta-heuristics are generally developed depending on some particular metaphors. These metaphor-based algorithms can be divided into groups based on the metaphors on which the methods are developed. These are evolutionary algorithms, swarm intelligence based algorithms, physics-based algorithms, immune algorithms, human-based algorithm, etc.

The working procedure of most of the algorithms is the same. These algorithms randomly generate some initial candidate solutions within the specific range and upgrade the solutions depending upon the metaphors they follow. In the process of up-gradation, each candidate solution upgrades itself locally and globally, which means the candidate explores new regions in the search space and exploit the region where they already obtained a potential solution. To develop an efficient meta-heuristic algorithm, proper balancing of exploration and exploitation is necessary. This is why the optimization community is developing hybrid methods to get a proper balance between these two characteristics. It has already been proved that, in many a case, a combination of two algorithms (maybe two meta-heuristics or parts of two meta-heuristics or one meta-heuristic combined with

another heuristic or any other method) gives better results in solving optimization problem than the component algorithms. For the last two decades, hybrid meta-heuristic algorithms have been gaining popularity, and scientists are widely using these methods to solve real-life combinatorial and other complicated optimization problems.

Metaheuristics have wide applications in the field of structural engineering. Structural problems are non-linear and multi-modal under some complex constraints. To eliminate these complexities, many researchers are using metaheuristic algorithms to solve such types of problems. Dede [1] used JAYA algorithm to solve the size optimization problem for steel grillage structures. Baradaran and Madhkhan [2] used an improved genetic algorithm (GA) for the optimal design of planar steel frames, Kaveh and Farhadmanesh [3] employed colliding body optimization (CBO) and enhanced colliding body optimization (ECBO) for optimal seismic design of steel plate shear walls, Kaveh et al. [4] proposed modified dolphin monitoring operator for weight optimization of skeletal structures. Teaching-learning based optimization (TLBO), ECBO, CBO, vibrating particles system (VPS), and harmony search algorithm (HS) were used by Kaveh et al. [5] for the optimal design of multi-span pitched roof frames with tapered members.

Considering the huge ability of meta-heuristic algorithms, researchers use these algorithms to optimize structural problems like retaining wall problems in recent decades. For instance, particle swarm optimization (PSO) was used by Varae [6] to optimize the cost and weight of a 4.5 m-high retaining wall. Again PSO with the passive congregation was utilized by Khajehzadeh et al. [7] in finding optimum cost design for 3 m and 5.5 m high retaining walls. Also, Khajehzadeh et al. [8] used modified particle swarm optimization (MPSO) in optimizing the cost of a 3 m-high retaining wall and sensitivity analysis of soil properties for a 4.5 m tall retaining wall. Gravitational search algorithm (GSA) was opted for finding the best cost design of 3 m and 5.5 m high retaining wall by Khajehzadeh and Eslami [9]. Yepes et al. [10] used simulated annealing (SA) for analyzing parameters of 4 m to 10 m retaining walls by considering soil properties variation. Ant colony optimization (ACO) was used by Bonab [11] for designing 3 m, 4 m, and 5 m-high retaining walls. Kaveh and Abadi [12] found out the optimal design of a 6.1 m high retaining wall by adopted harmony search (AHS), while the charged system search algorithm (CSS) was used for the same wall by Kaveh and Behnam [13]. Big bang big crunch (BBBC) was used by Camp and Akin [14] for optimizing cost and weight of 3 m-high and 4.5 m-high retaining walls, including a base shear key effect. The same problem was also solved by Gandomi et al. [15] by exploring swarm intelligence techniques efficiency with the same case studies. Nama et al. [21] incorporated HS, PSO, and TLBO for optimizing seismic active earth pressure

coefficient. Optimum design of the tied-back retaining wall was accomplished by Jasim and Al-Yaqoobi [17] using GA. Differential evolution (DE) was used for the optimal design of the cantilever retaining wall by Kumar and Suribabu [18]. For retaining wall optimization, Bath et al. [19] used DE, and PSO was used for optimum design of reinforced concrete retaining walls by Moayyeri et al. [20]. Nama et al. [21] used an improved backtracking search algorithm (IBSA) to obtain the active earth pressure on retaining wall supporting c - ϕ backfill using the pseudo-dynamic method. Except for all the above cases, meta-heuristic algorithms were used for many applications in structural engineering. Some of those can be seen in [22–32].

The BOA [33] is a nature-based algorithm that mimics butterflies' food and mating pair search behaviors. The system is focused primarily on the butterflies' foraging technique, which uses their smelling sense to assess the nectar or mating partner. The global phase of BOA uses the best butterfly's information to forage, and the local phase uses a random search or explores the domain for potential solutions. Both these phases make the BOA a suitable algorithm to apply in the different optimization problems. Although the literature indicates enough scope for the BOA to explore the search space, it faces some problems such as local optima stagnation, slow convergence, and skipping real solutions when solving real-life problems. That is why many articles have come up in the literature, improving the basic BOA so that the algorithm can perform better than the basic BOA. Some of the improved or modified algorithms of BOA can be found in [35]. Also, some of the hybrid methods where BOA is a component algorithm can be seen in [36–38].

The SOS [39] algorithm is implemented through the ecosystem's interactive behavior of different organisms. SOS is considered one of the most common algorithms for solving complex real-world problems because of its simplicity and higher proficiency. For various optimization problems of different disciplines, SOS has already proven its effectiveness. The stages of mutualism and commensalism make good use of population data as these two stages use the position of its best organism as a point of reference, helping to exploit the possible solutions. Again, by changing the current solution to generate new solutions, the parasitism process eliminates the inferior solutions and helps to explore the entire search room, allowing the algorithm to be well balanced in both exploratory and exploitative competencies. The simplicity of the SOS makes it one of the most competitive swarm intelligence based algorithm. Motivated by the above claims, we believe that the effectiveness of SOS must be further improved by hybridizing with other efficient algorithms to make the algorithm more efficient, stable, and robust. Numerous works on SOS in literature have tried to improve the algorithm by modifying it in various manners to improve its performance. Some of the works on improving

and hybridizing SOS can be found in [40–43]. For detailed improvement, hybridization, and applications of SOS, one can see [44].

The present study suggests a hybrid meta-heuristic, viz. h-BOASOS, combining BOA and SOS and applied it to solve the cantilever retaining wall problem. It is seen that the strength of BOA lies in its exploration capability whereas, SOS is known for its better exploitation capacities. To balance the exploration and exploitation of the algorithm, the search agents first search the entire region through the steps of BOA to discover the possible solutions throughout the entire search space. Once the possible solutions are obtained, the solutions are passed through the SOS steps to take advantage of the search space to find the optimal solution near the potential solutions already obtained by BOA. Thus the suggested h-BOASOS is assumed to be a well-balanced and robust algorithm that can find the optimal global solution. By applying the algorithm to solve a suite of 35 (thirty-five) benchmark functions of varying complexities, the efficacy of the proposed h-BOASOS was assessed and the results obtained are contrasted with 10 (ten) state-of-the-art algorithms. To prevent the error due to the algorithm's stochastic nature, each function was executed for 30 runs with 10,000 iterations. For analysis, mean and standard deviation values are reported. The experimental results indicate that the algorithm proposed is more potent than those of the algorithms compared to h-BOASOS. For further study, the non-parametric Friedman rank test has been performed considering the mean values of the proposed and compared algorithms and found that the rank of h-BOASOS is the least among all compared algorithms. For all comparative algorithms, convergence graphs of selected functions are plotted to scrutinize the convergence velocity and find that the convergence rate of the proposed algorithm is more robust than that of other algorithms. It is also applied to solve 04 (four) real-world engineering design problems, namely, gear train design problem, cantilever beam design problem, car-side impact design problem, and three-bar truss design problem to assess the problem-solving skill of the h-BOASOS. The results obtained to support our assertion of h-BOASOS' supremacy and argue that the proposed algorithm is better than some of the literature's well-known meta-heuristics. Finally, h-BOASOS was used to obtain the optimum cost and weight of the cantilever retaining wall problem, a relatively complex problem with 12 (twelve) design variables and 16 (sixteen) constraints.

The remaining part of paper has been organized as follows: the two component algorithms, BOA and SOS have been described in Sects. 2 and 3 respectively. The proposed algorithm has been elaborated in Sect. 4. Section 5 deals with the experimental setup. Section 6 describes the results of the proposed algorithm. Applications of this hybrid method in different real-life problems are described

in Sect. 7. Section 8 describes the cantilever retaining wall problem with its mathematical formulation. Section 9 discusses the results of the problem with the aid of the proposed h-BOASOS and some other methods. Finally the conclusion and future scope of this work has been presented in Sect. 10.

2 Butterfly optimization algorithm

Butterfly optimization algorithm (BOA) was introduced by Arora and Singh [33] in 2018, mimicking the food foraging and mating pair behavior of butterflies. In BOA, butterflies are used as search agents for the algorithm as BOA is a population-based iterative algorithm. The fragrance generated by a butterfly is propagated to all other butterflies in the region, and other butterflies sense the propagated fragrance through chemoreceptors scattered over the body of butterflies, which are used to smell the fragrance. Thus a collective social knowledge network is formed in BOA, which helped the algorithm to search the entire space effectively. The candidate butterflies are always moving towards the best butterfly once they smell the best butterfly's fragrance. This is known as the global search phase of BOA.

On the other hand, when a butterfly does not smell the fragrance from any butterfly, a random movement is considered, which is termed as a local search phase in BOA. The basic concept of sense depends on three special parameters viz., sensory modality (c), stimulus intensity (I), and power exponent (a). In BOA, it is assumed that many butterflies release fragrance simultaneously, and by the sensory modality, butterflies sense and differentiate the fragrances. I characterizes the fitness of a butterfly in BOA, and a is the response compression. The behavior of butterflies depends on two imperative factors viz. variation of I , and formulation of f . In BOA, the stimulus intensity I of a butterfly is linked with the value of the objective function, and f is relative. Considering all these aspects, the fragrance is formulated in the following way:

$$f_i = c * I^a \quad (1)$$

where, f_i is the apparent magnitude of the fragrance of i th butterfly.

It has been already discussed that BOA consists of two crucial phases, viz., global search phase and local search phase. In the global phase, a butterfly moves towards the best butterfly in a particular iteration. Mathematically, it is represented as

$$X_i^{new} = X_i + (r^2 * g^{best} - X_i) * f_i \quad (2)$$

where, X_i is the i th butterfly and g^{best} represents the current best solution in a particular stage. The fragrance of i th butterfly is represented by f_i and r is a random number in $[0, 1]$.

The local search phase of BOA can be represented as

$$X_i^{new} = X_i + (r^2 * X_j - X_k) * f_i \quad (3)$$

where, X_j and X_k are the j th and k th butterflies from the solution space and selected randomly from the current population, and r is a random number in between 0 and 1.

3 Symbiosis organisms search

In 2014, Cheng and Prayogo [39] proposed SOS algorithm mimicking the interactive relationship among the organisms in an ecosystem. The SOS is executed in three steps: mutualism phase, commensalism phase, and parasitism phase. During the optimization process, a set of initial solution is generated randomly within the search space, and each solution is updated by the three phases. The details of SOS are summarized below:

3.1 Mutualism phase

In the mutualism phase, two organisms of the ecosystem interact mutually, and both get benefits from the interaction. An organism X_i tries to interact with an organism X_j , where X_j is randomly selected from the ecosystem. From the interaction, both the organisms try to improve their mutual survival benefit in the ecosystem. Mathematically, the new X_i and X_j are determined by Eqs. (4) and (5) and update in the ecosystem. The new organism will replace the old one if the new organism is better than its previous fitness value.

$$X_i^{new} = X_i + r_1 * (X_{best} - Mutual_{Vector} * BF_1) \quad (4)$$

$$X_j^{new} = X_j + r_2 * (X_{best} - Mutual_{Vector} * BF_2) \quad (5)$$

where,

$$Mutual_{Vector} = (X_i + X_j)/2 \quad (6)$$

Here, X_{best} represents the best organism in the ecosystem, the value of benefit factors (BF_1 and BF_2) are considered randomly as either 1 or 2. These factors represent the level of benefit to each organism, i.e., whether an organism partially or fully benefits from the interaction. ' $Mutual_{Vector}$ ' represents the characteristic of the relationship of the organism X_i and X_j . r_1 and r_2 are two random numbers between 0 and 1.

3.2 Commensalism phase

When, in the interaction of two organisms, one gets benefited, and the other neither benefited nor harmed, this interaction is called commensalism. In the commensalism phase of SOS, an organism X_i increases its beneficial advantage by

interacting with another organism X_j to a higher degree of adaptation in the ecosystem, but X_j remains the same. The new fitness value of X_i is determined by Eq. (7) and updates the new X_i if its fitness value is better than the previous X_i .

$$X_i^{new} = X_i + r_3 * (X_{best} - X_i) \quad (7)$$

where, X_{best} is the best organism according to the fitness value in the ecosystem and r_3 is any random number between -1 and 1 .

3.3 Parasitism phase

On the interaction of two organisms, if one organism gets benefited but the other organism gets harmed, then this interaction is called a parasitism relationship in an ecosystem. In the SOS algorithm, a parasite vector, namely $Parasite_{vector}$ is determined by duplicating an organism X_i . The $Parasite_{vector}$ is obtained in the following way:

- Select one dimension randomly from the dimensions of organism X_i i.e. from $1, 2, 3, \dots, D$.
- Randomly modify the selected dimension using Eq. (8) and replace the position of the dimension into X_i .

$$X_{i,d} = l_d + r_4 * (u_d - l_d) \quad (8)$$

Where, r_4 is a random number between 0 and 1 .

If the objective function value of the $Parasite_{vector}$ is superior to that of X_j , $Parasite_{vector}$ will replace X_j in the ecosystem. If the reverse happens, X_j is sustained and the $Parasite_{vector}$ leaves the ecosystem.

4 Proposed methodology

An efficient and robust algorithm can be developed only when the balance between exploring new areas in search space and exploiting prior knowledge that has been already gathered in the search process is assured. When exploration is better than exploitation, diversification becomes poor and causes premature convergence. On the other hand, when exploitation is better than exploration, it causes poor intensification, and thus the algorithm's convergence rate becomes slower. The proposed meta-heuristic seems to have a brilliant combination of exploration and exploitation using both the global and local phases of BOA together with mutualism, commensalism, and parasitism steps of SOS. The proposed h-BOASOS seems to have the co-ordination of exploring the global search process, together with the exploitation of the search process.

BOA uses a random number r in order to update the location of butterflies. The choice of the search process of butterflies depends on a factor known as switch probability. In

the original BOA, the switch probability (p) was taken as 0.8 . So, when the value of random number r is less than 0.8 , the butterflies move into the global search phase, indicating that most butterflies move towards the global phase when a smaller number of butterflies choose the local search phase. This makes the algorithm weaker in exploration in the search space though it is relatively more robust in exploiting the search process.

On the other hand, SOS is known for its stronger local searchability. The mutualism and commensalism phases concentrate on generating new organisms and select the best organism for survival. Provided that a new solution is updated relied on the best organism, the convergence rate often speeds up; however, local solutions may occur. The mathematical equations of these two phases show that the algorithm allows the organisms to search in the domain to discover near the best solutions they already obtained in the search space, thereby improving the exploitation ability of the algorithm. Again, the parasitism phase enables the search process to avoid the solution from trapping in local optima, thus improving the exploration ability of the algorithm. The improvements happen because a new candidate solution, namely, $Parasite_{vector}$ is generated in the ecosystem by replacing some random elements of the organism X_i with other elements. Then $Parasite_{vector}$ is compared to another randomly selected organism in the ecosystem. Also, in the mutualism phase, two organisms update their mutual existence from their interaction and the same organisms that already updated their position through mutualism again update their positions to attain the global optima in the commensalism phase. These double impacts help the algorithm to expedite the convergence rate and make the algorithm competitive than other meta-heuristic algorithms.

From the above observations, to overcome the lacunae of both the algorithms, we have tried to construct a better algorithm by combining the above two algorithms BOA and SOS to propose a novel hybrid algorithm for optimization and named it h-BOASOS. The descriptions of BOA and SOS has already been given in Sects. 2 and 3 for convenience.

The main objective of the proposed algorithm is to combine the exploration and exploitation skill of SOS with the exploration facility of BOA. The original SOS has great exploitation strength, and the reason behind it is already explained above, whereas, in contrast, the basic BOA has very efficient exploration capability as the majority of the butterflies in the swarm choose the global phase of the algorithm. Thus, the proposed h-BOASOS algorithm owns an excellent convergence speed, can come out from the possible entrapment in the local optima, and possesses an excellent computational ability. Pseudo-code and flowchart of the proposed h-BOASOS are given in Table 1 and Fig. 1 respectively.

Table 1 Pseudo-code of the proposed h-BOASOS algorithm

Input:
 Objective function $f(X)$, $X = (X_1, X_2, \dots, X_{dim})$, where dim is the total number of dimensions;
 Max_{iter} : maximum number of iteration
%Initialization%
 Set Initial generation/iteration number $G = 0$; Generate initial population of n Butterflies $X_i (i = 1, 2, \dots, n)$;
 Stimulus Intensity I_i of X_i is determined by $f(x_i)$.
 Initialize the parameters: **BF1** (Benefit Factor 1), **BF2** (Benefit Factor 2), **C** (sensor modality), **a** (power exponent),
P (switch probability), and $Mutual_{vector}$
While stopping criteria do not met **do**
 For each butterfly in the population **do**
 Calculate fragrance for each butterflies using Eq. (1)
 End For
 Find the best solution;
 For each butterfly x_i in the population **do**
 Generate a random number r from $[0,1]$
 If $r < p$ then
%Global Phase%
 Move towards best butterfly using Eq. (2)
 Else
%Local Phase%
 Move randomly using Eq. (4)
 End If
 Calculate fitness value of the new Butterfly and update;
% Mutualism Phase%
 Randomly select one Butterfly $x_j (i \neq j)$;
 Evaluate the mutual relationship vector of x_i and x_j using Eq. (3);
 Update population x_i and x_j using Eqs. (5) & (6);
 Calculate the fitness of x_i and x_j ;
Commensalism Phase
 Randomly select one population $x_j (i \neq j)$;
 Update x_i with the aid of x_j using Eq. (7);
 Calculate the fitness of pop_i ;
Parasitism Phase
 Make a parasite vector by firstly duplicating pop_i and then randomly updating the dimensions;
 Evaluate the fitness of the $parasite_{vector}$ and compare with the host;
 End For
End While
 Output: The best solution found.

4.1 Parameters used in the proposed h-BOASOS

In the proposed hybrid h-BOASOS, the parameters of both the component algorithms, BOA and SOS are incorporated. The value of the parameters are given in Table 2.

4.2 Iterative technique used in proposed h-BOASOS

The newly suggested h-BOASOS combines the concepts of BOA and SOS and generates solutions in every new generation by using different operators and mechanisms of BOA

and SOS. The steps of the h-BOASOS can be described as follows:

Step 1: Generate the initial candidate solutions randomly to create an initial population.

Step 2: For each candidate solution, the fragrance is generated using the formulation as a function of the physical intensity of stimulus using Eq. 1.

Step 3: For the iterative phase, one random number r is generated and checked with the switching probability (p).

Step 4: If the global exploration has opted, the candidate solution gets updated using Eq. 2.

Fig. 1 Flowchart diagram of the proposed h-BOASOS

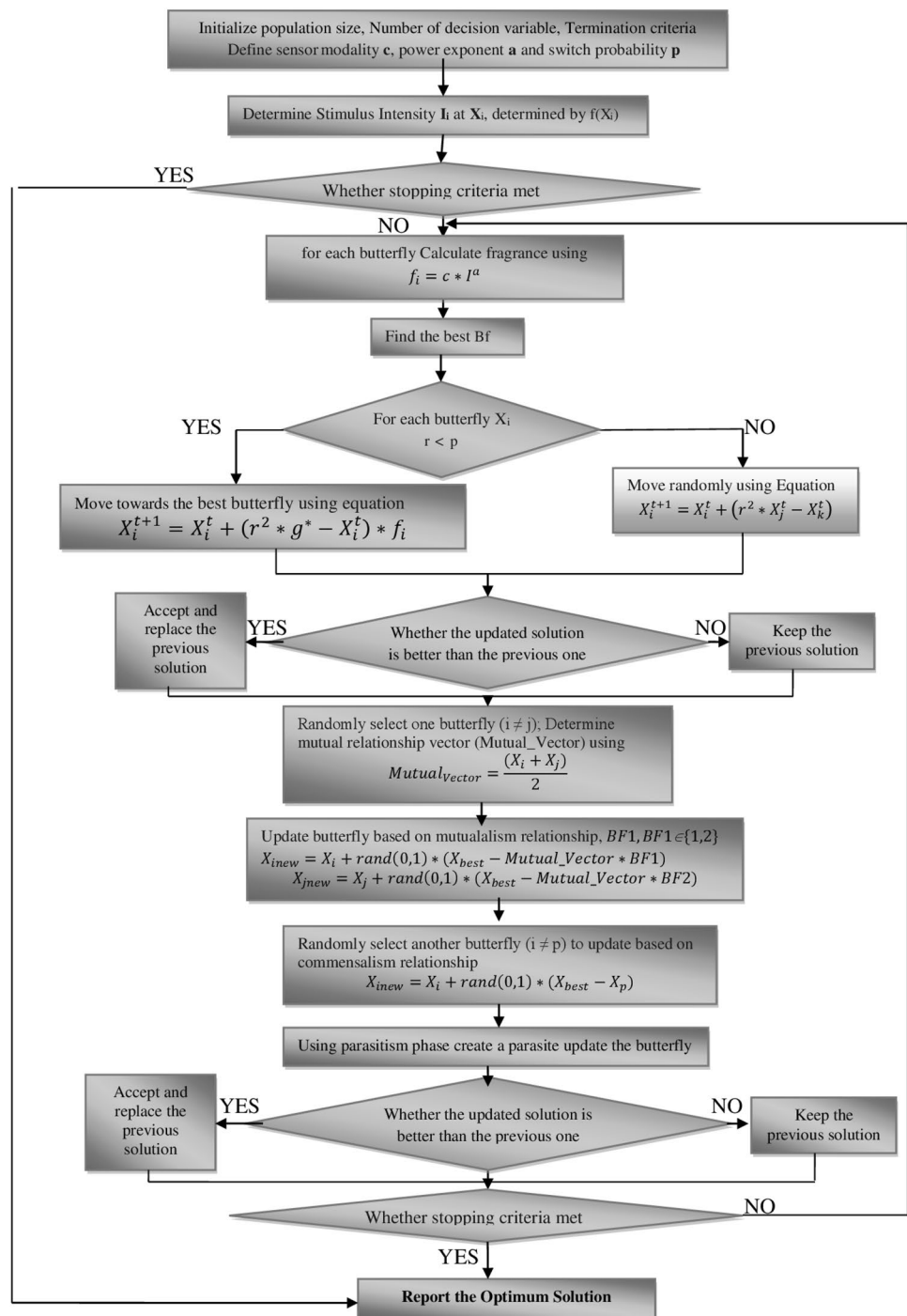


Table 2 Values of parameters of the proposed algorithms

Parameter	Value
Population size (NP)	50
Modular modality (C)	0.01
Power exponent (a)	0.1–0.3 (Increases uniformly)
Switch probability (p)	0.8
BF1	1 or 2 (Randomly)
BF2	1 or 2 (Randomly)

Step 5: If the local phase is selected, the candidate solution takes a random search and updates using Eq. 3.

Step 6: The fitness value of each candidate solution is checked with the previous one and replaced if needed.

Step 7: The updated solution goes for the mutualism phase by choosing a random solution from the search space and find the mutual Vector using Eq. 6.

Step 8: With the mutual Vector, both of the candidate solutions are updated by using Eqs. 4 and 5.

Step 9: In the commensalism phase, the opted candidate solution is updated using Eq. 7.

Step 10: In the parasitism phase, one $Parasite_{vector}$ is formed and checked with the recent candidate solution and replaced if needed.

Step 11: The fitness of all the candidate solution is obtained and checked with the fitness of the previous one.

Step 12: The approach is stopped when the termination condition (maximum number of iterations) has been reached.

5 Experimental setup

In this section, the experimental set-up that we have performed to show the proposed algorithm’s superiority is discussed briefly.

To check the efficiency, the h-BOASOS algorithm is tested on 35 different benchmark functions of varying complexity as shown in Tables 3 and 4. These benchmark functions are chosen to examine various features of the algorithm such as convergence rate, attainment of a large number of optimal points, ability to jump out of local optima, and avoid premature convergence. The stopping criteria can be defined in different ways like maximum CPU time used, maximum iteration number reached, the maximum number of iterations with no improvement, a particular value of the error rate, or any other appropriate criteria. In our case, we have taken the maximum number of iteration as the stopping criterion.

6 Results and discussions

6.1 Comparison with some state-of-the-art algorithms

To measure the performance of a meta-heuristic, several tests should be conducted to approve an algorithm’s performance. As these algorithms are stochastic in nature, it should be carefully noticed that an appropriate and a sufficient number of test functions and case studies are engaged to assure that the obtained results are occurring through its solid mathematical base and not by chance.

In the present study, a set of 35 classical benchmark functions of varied complexities are considered from literature to examine the competence of the proposed h-BOASOS, and the performance is compared with some state-of-the-art meta-heuristics, including BOA and SOS. Knowing the meta-heuristics’s stochastic nature, each of the algorithms is run 30 times to avoid the variability of the results. For this comparative analysis, the population size is fixed at 50; the maximum number of iterations is taken as 10,000. The performance of h-BOASOS over the aforesaid benchmark functions is compared with 12 (twelve) state-of-the-art algorithms: two component algorithms (BOA and SOS) and 10 (ten) popular meta-heuristics DE [45], PSO [46] and JAYA [48] sine cosine algorithm (SCA) [49], artificial bee colony ABC [47], cuckoo search (CS) [51], firefly algorithm (FA) [52], GA [53], monarch butterfly optimization (MBO) [54], and moth flame optimization (MFO) [50]. To compare the performances of the algorithms, simulation results containing the average

Table 3 Unimodal classical benchmark functions

Function ID	Name	Formulation of problem	Dim	Range	f_{min}
f_2	Cigar	$f(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$	30	[-100, 100]	0
f_3	Step	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$	30	[-100, 100]	0
f_4	Quadratic	$f(x) = \sum_{i=1}^D i * x_i^4 + random(0, 1)$	30	[-1.28, 1.28]	0
f_{12}	Rosenbrock	$f(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
f_{13}	Schwefel 1.2	$f(x) = \sum_{i=1}^D \sum_{j=1}^i x_j^2$	30	[-100, 100]	0
f_{14}	Schwefel 2.21	$f(x) = max x_i , 1 \leq i \leq D$	30	[-100, 100]	0
f_{15}	Schwefel 2.22	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30	[-10, 10]	0
f_{18}	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[-10, 10]	0
f_{19}	Powell	$f(x) = \sum_{i=1}^{D/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-3} + 10x_{4i-2})^2 + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	30	[- 100, 100]	0
f_{20}	Sum Square	$f(x) = \sum_{i=1}^D x_i^2 * i$	30	[-10, 10]	0
f_{21}	Zettl	$f(x) = (x - 1^2 + x - 2^2 - 2x_1)^2 + 0.25x_1$	2	[-1, 5]	- 0.0003075
f_{26}	Brown	$f(x) = \sum_{i=1}^{D-1} (x_i)^{2x_{i+1}^2+1} + (x_{i+1})^{2x_i^2+1}$	30	[-1, 4]	0
f_{32}	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[-1010]	0
f_{34}	Dejong5	$f(x) = (0.002 + \sum_{i=1}^{25} \frac{1}{i+(x_1-a_{1i})^6+(x_2-a_{2i})^6})^{-1}$	2	[-65.536, 65.536]	
f_{35}	Tablet	$f(x) = (10^6)(x_1^2) + \sum_{i=2}^D x_i^6$	2	[-100100]	

Table 4 Multimodal classical benchmark functions

Function ID	Name	Formulation of problem	Dim	Range	f_{min}
f_1	Sphere	$f(x) = \sum_{i=1}^D x_i^2$	30	[-100, 100]	0
f_5	Bohachevsky	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	[-100, 100]	0
f_6	Ackley	$f(x) = 20 + e - 20e^{1/D(\sqrt{1/D \sum_{i=1}^D x_i^2})}$	30	[-100, 100]	0
f_7	Griewank	$f(x) = \sum_{i=1}^D x_i^2 / 4000 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) - 1$	30	[-100, 100]	0
f_8	Levy	$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3x_2)] + (x_1 - 1)^2 \sin^2(2\pi x_2)$	2	[-10, 10]	0
f_9	Rastrigin	$f(x) = 10D + \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i)]$	30	[-5.12, 5.12]	0
f_{10}	Alpine	$f(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	30	[-10, 10]	0
f_{11}	Schaffer	$f(x) = (x_1^2 + x_2^2)^0.25 [50(x_1^2 + x_2^2)^0.1 + 1]$	2	[-100, 100]	0
f_{16}	Schwefel 2.26	$f(x) = -\sum_{i=1}^D [x_i \sin(\sqrt{ x_i })]$	30	[-500, 500]	-418.982
f_{17}	Goldstein price	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * \dots [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
f_{22}	Foxholes	$f(x) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^D x_i - a_j^6}]^{-1}$	2	[-65.536, 65.5360]	0.0003075
f_{23}	Branin	$f(x)(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	[[−50], [1015]]	0.3978873
f_{24}	Kowalik	$f(x) = \sum_{i=1}^{11} [a_i - x_1(b_i^2 + b_i x_2)] / b_i^2 + b_i x_3 + x_4]^2$	30	[-50, 50]	0
f_{25}	elliptic	$f(x) = \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1}) + (y_D - 1)^2] \dots + \sum_{i=1}^D u(x_i, 10, 100, 4) + \pi / D 10 \sin^2(\pi y_1)$	30	[-50, 50]	0
f_{27}	Levy N.13	$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	[-10, 10]	0
f_{28}	Scaffer 4	$f(x) = \sum_{i=1}^D \sum_{j=1}^5 j \cos((j+1)x_i + j)$	2	[-10, 10]	-25.740858
f_{29}	Bohachevsky2	$f(x) = x_1^2 + 2x_2 - 2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.3$	2	[-100, 100]	0
f_{30}	Bohachevsky3	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) + 4\pi x_2 + 0.3$	2	[-100, 100]	0
f_{31}	Perm $func$ 0, d, β	$f(x) = \sum_{i=1}^D (\sum_{j=1}^D (j + \beta)(x_j^i - \frac{1}{j})^2)$	30	[-30, 30]	0
f_{33}	Dixon Price	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$	30	[-10, 10]	0

results (MEAN) and the corresponding standard deviations (STD) are recorded for each algorithm. Firstly, the results of h-BOASOS and six other algorithms (BOA, SOS, DE, PSO, JAYA, and SCA) are recorded in Tables 5 and 6. Secondly, the newly suggested algorithm is compared with six other meta-heuristics (ABC, CS, FA, GA, MBO, and MFO), and the results are shown in Tables 7 and 8.

From the simulation results shown in Table 5, it can be interpreted that the performance of the proposed h-BOASOS is better than the basic BOA and SOS in terms of searching function minimum. From this table, it is found that h-BOASOS is superior to BOA and SOS in 17 and 16 benchmark functions, similar to 16 and 12 functions, and inferior in 2 and 7 occasions, respectively.

To evaluate the performance of proposed h-BOASOS, it is compared to ten other state-of-the-art algorithms, which are the best performing and produce a satisfactory performance when applied to global optimization problems. These algorithms are widely employed to compare the performance of optimization algorithms, and the results are recorded in Tables 5 and 7.

The observation can be made from Table 5, h-BOASOS obtained superior solution than DE, PSO, JAYA, and SCA in 23, 29, 24, and 33 functions, similar in 5, 3, 3, 1 cases and inferior in 7, 3, 6, 1 occasions.

By Table 7, it can be evaluated that h-BOASOS generate superior solution compared to ABC, CS, FA, GA, MBO, and MFO in 28, 33, 22, 28, 29, and 15 benchmark functions. h-BOASOS obtained similar results with the said algorithms in 2, 1, 7, 2, 3, and 19 functions. Again, h-BOASOS is inferior to the algorithms in 5, 1, 6, 5, 3, and 1 occasions (Tables 9, 10).

6.2 Statistical rank test

Also, to evaluate the statistical performance of the newly developed algorithm, we have introduced the Friedman rank test, a non-parametric statistical test considering all the algorithms and mean values found by the respective algorithms found on the Tables 5, 6, 7 and 8. The mean rank found for all the algorithms after performing the Friedman’s test is given in Tables 11, 12. From these two tables, it is evident

Table 5 Simulation results of h-BOASOS and other state-of-the-art algorithms for functions 1 to 28 with NP=50 and 30 execution each with 10,000 iterations

Function ID		h-BOASOS	BOA	SOS	DE	PSO	JAYA	SCA
1	MEAN	0	0	6.03E-173	2.62E-165	1.77E-51	2.68E-39	0
	STD	0	0	0	0	1.12E-50	3.40E-39	4.84E-04
2	MEAN	0	0	0	1.41E-172	3.10E-53	1.97E-26	1.53E+05
	STD	0	0	0	0	2.11E-52	1.60E-26	2.24E+05
3	MEAN	0	0	0	1.30E-01	1.31E-31	0	2.00E-04
	STD	0	0	0	4.85E-01	6.65E-31	0	1.00E-04
4	MEAN	1.33	3.89E-05	1.74	2.62E-03	3.74	2.10	0
	STD	1.48E-01	2.90E-05	2.05E-01	1.05E-03	1.03	2.55E-01	1.40E-03
5	MEAN	0	0	0	0	3.04E-02	0	4.33E-03
	STD	0	0	0	0	2.62E-17	0	1.90E-02
6	MEAN	8.88E-16	1.72	1.00E-15	1.72	1.82E+01	4.44E-15	3.85E-03
	STD	0	0	6.38E-16	0	5.48	0	1.00E-02
7	MEAN	0	1.85E-19	0	9.22E-17	4.05E+01	4.47E-01	1.74E-01
	STD	0	2.69E-20	0	2.05E-17	4.13E+01	7.83E-02	2.12E-01
8	MEAN	3.72E-04	4.41E-01	6.73E-23	5.98E-02	4.93	1.35E-31	1.17
	STD	6.09E-04	5.75E-02	1.95E-22	2.08E-01	6.00	6.68E-47	2.29E-01
9	MEAN	0	0	0	1.25E+01	1.45E+02	2.89	1.50E-02
	STD	0	0	0	3.31	3.80E+01	3.31	4.04E-02
10	MEAN	0	0	0	6.90E-16	4.44E-02	4.52E-08	1.02E-10
	STD	0	0	0	2.58E-15	4.53E-01	1.33E-07	4.19E-10
11	MEAN	0	0	4.85E-16	0	0	5.73E-11	2.00E-08
	STD	0	0	2.74E-16	0	0	3.87E-11	5.33E-08
12	MEAN	8.48	2.88E+01	1.82E-01	2.36E+01	2.57E+02	5.17E-01	2.88E+01
	STD	5.58E-01	3.13E-02	1.20E-01	4.99	3.73E+02	9.01E-01	3.33E-02
13	MEAN	0	0	0	1.14E-175	1.20E+03	1.49E-38	4.52E+04
	STD	0	0	0	0	5.45E+02	1.68E-38	3.07E+04
14	MEAN	0	7.00E-153	2.27E-74	1.02	1.06E-01	2.67E-09	3.71
	STD	0	1.48E-152	2.41E-74	4.60E-01	6.54E-02	2.47E-09	1.78
15	MEAN	0	0	1.24E-86	8.15E-91	7.28E+01	2.68E-21	3.20E-03
	STD	0	0	1.26E-86	1.23E-90	2.27E+01	2.01E-21	7.59E-03
16	MEAN	- 6.52E+03	- 2.27E+03	- 4.19E+02	- 4.57E+12	- 7.57E+03	- 5.11E+03	- 4.19E+02
	STD	8.06E+02	4.56E+02	0	3.19E+12	8.89E+02	3.48E+02	0
17	MEAN	3.00	3.00	3.00	3.00	3.00	3.00	3.00
	STD	6.43E-06	0	1.94E-15	0	0	8.46E-04	1.19E-05
18	MEAN	0	0	2.62E-43	0	0	3.70E-21	0
	STD	0	0	5.87E-43	0	0	1.42E-20	0
19	MEAN	0	0	7.51E-13	1.31E+02	3.37E+03	1.10E-08	7.73E-05
	STD	0	0	3.99E-12	1.85E+02	9.52E+02	1.08E-08	3.48E-04
20	MEAN	0	0	1.09E-117	1.46E-19	9.20E+02	2.43E-81	1.40E+01
	STD	0	0	3.94E-117	1.18E-19	8.27E+02	1.26E-80	4.05
21	MEAN	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.52E-03
	STD	1.29E-05	0	1.76E-18	4.57E-19	0	8.05E-09	1.03E-03
22	MEAN	9.98E-01	2.84	9.98E-01	9.98E-01	1.20	9.99E-01	1.61
	STD	1.13E-03	2.55	2.92E-16	0	9.12E-01	2.87E-03	1.15
23	MEAN	3.98E-01	4.98E-01	3.98E-01	3.98E-01	3.98E-01	3.99E-01	3.98E-01
	STD	1.58E-05	1.47E-01	3.17E-07	0	5.95E-08	1.60E-03	5.94E-07
24	MEAN	3.73E-04	6.66E-04	3.27E-04	3.07E-04	4.05E-04	3.72E-04	4.67E-04
	STD	9.98E-05	7.52E-04	7.60E-05	1.25E-08	2.44E-04	1.38E-04	2.50E-04

Table 5 (continued)

Function ID		h-BOASOS	BOA	SOS	DE	PSO	JAYA	SCA
25	MEAN	0	0	0	1.99E-153	1.40E-08	3.70E+03	2.30E+03
	STD	0	0	0	2.64E-153	2.20E-08	1.09E+03	6.19E+03
26	MEAN	0	0	0	2.26E-154	1.17E-13	9.17E-03	5.74E-06
	STD	0	0	0	1.64E-154	2.10E-13	3.30E-03	1.28E-05
27	MEAN	0	1.23	1.35E-31	2.72E-19	1.41	1.35E-31	9.28E-04
	STD	0	7.74E-01	6.68E-47	4.72E-19	1.80	6.68E-47	2.57E-03
28	MEAN	0	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01
	STD	0	1.36E-05	0	2.85E-06	3.39E-05	3.65E-06	0

Table 6 Simulation results of h-BOASOS and other state-of-the-art algorithms for functions 29 to 35 with NP=50 and 30 execution each with 10,000 iterations

Function ID		h-BOASOS	BOA	SOS	DE	PSO	JAYA	SCA
29	MEAN	0	0	0	2.01E-07	5.29E+01	0	3.13E-03
	STD	0	0	0	5.41E-07	8.74E+01	0	9.54E-03
30	MEAN	0	0	0	4.45E-05	4.02E+01	2.04E-17	1.60E-04
	STD	0	0	0	6.10E-05	8.14E+01	1.11E-16	5.14E-04
31	MEAN	0	8.79E+03	1.13E-05	1.01E+58	5.49E+21	1.71E+02	2.51E+76
	STD	0	5.57E-12	1.57E+02	3.11E+58	3.01E+22	2.70E+02	1.17E+77
32	MEAN	0	2.07E-02	0	7.10E-05	1.58	0	1.34E-03
	STD	0	2.46E-02	0	8.69E-05	2.98	0	2.63E-03
33	MEAN	0	9.86E-01	6.67E-01	2.46E+03	1.66E+01	6.45E-01	3.19E+03
	STD	0	1.45E-03	1.41E-16	7.23E+02	1.18E+01	1.22E-01	1.70E+04
34	MEAN	0	1.75	9.98E-01	9.98E-01	9.17	9.98E-01	1.30
	STD	0	9.07E-01	1.13E-16	1.27E-16	8.56	4.36E-04	7.02E-01
35	MEAN	0	0	0	5.29E-306	1.19E-13	1.19E+05	2.50E+09
	STD	0	0	0	0	6.08E-13	1.77E+05	9.48E+09

that the rank of the proposed hybrid h-BOASOS is minimum in both cases. Table 6 shows that the algorithms’ respective ranks are 1 for h-BOASOS with mean rank 2.29, 2 for SOS, 3 for BOA, 4 for DE, 5 for JAYA, and 6 for PSO from the Table 11. Also, from Table 12, it can be seen that the first mean rank holder is the proposed h-BOASOS, the second is ABC, third-GA, fourth-MBO, fifth-CS, sixth-MFO, and seventh is FA. Finally, an overall Friedman rank test is also performed considering all the compared algorithms, shown in Table 13. Evaluating Table 13, we can rank 1 for h-BOASOS with minimal mean rank (3.49), 2 for ABC with 3.78 mean rank, 3 for SOS with mean rank 4.40, followed by BOA, DE, JAYA, GA, MBO, CS, SCA, PSO, MFO, and FA. So it may be concluded that the proposed h-BOASOS is statistically superior to all of the compared algorithms.

6.3 Convergence analysis

To verify the convergence speed of h-BOASOS, some of the convergence graphs with its component algorithms BOA and SOS, as well as with other five well-known algorithms DE PSO, JAYA, SCA, and MFO are presented in Fig. 2a–f.

Different unimodal and multimodal functions for which convergence graphs are given here are Booth, Ackley, Branin, Dejong5, Griewank, Levy N.13, Schaffer 4, and Foxholes functions. The best candidate solution’s fitness value in each iteration is considered to draw convergence curves in these figures. From these figures, it can be concluded that the proposed h-BOASOS algorithm converges faster for both types of functions, which establishes that the exploration and exploitation capacities of the proposed algorithm are more balanced than the other chosen algorithms.

7 Real world problem

7.1 Gear train design problem

Gear Train Design Problem: The main objective of this engineering design problem is to minimize the gear ratio for a given set of four gears of a train. The number of teeth of the gears is the parameter here. It is an unconstrained problem,

Table 7 Simulation results of functions 1 to 27 of h-BOASOS and some other algorithms with NP=50 and 30 execution each with 10,000 iterations

f(x)		h-BOASOS	ABC	CS	FA	GA	MBO	MFO
1	MEAN	0	2.58E-01	6.49E-61	7.87E-04	2.20E-02	2.21E-01	0
	STD	0	5.13E-01	9.77E-61	1.37E-04	9.68E-03	1.41E-01	0
2	MEAN	0	8.43E-03	5.54E-63	1.08E-05	2.20E-04	3.40E-04	0
	STD	0	2.30E-02	8.93E-63	1.46E-06	1.10E-04	4.79E-04	0
3	MEAN	0	2.23E-01	1.22	0	2.30E-02	3.17E+04	1.13E-04
	STD	0	4.83E-01	1.49	0	1.05E-02	2.57E+04	9.87E-05
4	MEAN	7.9526	5.14E-01	1.24E+01	5.96E-02	2.74E-03	3.49E-03	9.12E-02
	STD	3.25E-01	1.31E-01	1.42	2.48E-02	8.51E-04	6.23E-03	4.64E-02
5	MEAN	0	0	4.92E-04	0	0	0	0
	STD	0	0	8.67E-04	0	0	0	0
6	MEAN	8.88E-16	3.65	1.71E+01	7.73E-04	4.67E-02	1.31E+01	8.88E-16
	STD	0	6.94E-01	5.46	4.81E-05	1.68E-02	9.42	0
7	MEAN	0	5.48E-01	2.01E+01	1.07E-07	5.56E-02	4.31E+02	0
	STD	0	2.72E-01	4.31E-02	1.31E-08	2.95E-02	1.65E+02	0
8	MEAN	4.47E-04	3.08E-02	1.62E+01	1.20	1.37E-04	3.16E+01	2.47
	STD	6.97E-04	4.82E-02	1.19E+01	1.67	7.66E-05	6.32E+01	3.68E-01
9	MEAN	0	2.36E+01	1.08E+02	3.47E+01	1.65E+01	6.38E+01	0
	STD	0	5.49	5.00E+01	9.55	4.11	4.30E+01	0
10	MEAN	0	6.06	1.06E+01	1.19E-02	3.15E-17	1.97E-01	0
	STD	0	6.16E-01	4.23	6.43E-03	3.08E-16	3.77E-01	0
11	MEAN	0	8.62E-16	9.82E-01	1.43E-02	1.69E-17	6.68E-04	0
	STD	0.854E-16	7.87E-01	5.35E-03	1.69E-16	8.09E-04	0	
12	MEAN	2.89E+01	1.25E+02	3.95E+01	1.84E+01	3.79E+01	7.14	2.89E+01
	STD	1.79E-01	8.77E+01	3.34E+01	4.24	2.53E+01	1.43E+01	6.80E-02
13	MEAN	0	2.37E-02	4.06E-02	5.24E-02	2.42E-03	1.12E+05	0
	STD	0	4.19E-02	6.06E-02	6.71E-02	1.07E-03	1.98E+04	0
14	MEAN	0	1.01E-01	3.67E-03	1.43E-04	4.67E-02	3.54	0
	STD	0	6.61E-01	1.58E-03	1.49E-05	1.09E-02	4.00	0
15	MEAN	0	1.43E-01	1.12E+01	1.50E-03	5.44E-02	7.02E-03	0
	STD	0	6.84E-02	2.50E+01	1.13E-04	1.62E-02	2.96E-03	0
16	MEAN	- 6.44E+03	- 1.05E+04	- 1.09E+02	- 7.15E+03	- 1.19E+04	- 3.94E+04	- 4.18E+02
	STD	7.77E+02	2.94E+02	1.08E+01	9.10E+02	2.95E+02	5.71E+04	2.47
17	MEAN	3.00	3.00	3.00	3.00	3.00	3.00	1.32E+01
	STD	4.91E-06	6.47E-12	0	0	0	2.96E-03	1.83E+01
18	MEAN	0	1.19E-04	1.33E-06	0	8.39E-104	0	0
	STD	0	1.30E-04	2.16E-06	0	2.52E-103	0	0
19	MEAN	0	1.52E+01	6.64	2.40E-04	3.27E-02	3.44E+03	0
	STD	0	1.11E+01	7.00	4.76E-05	1.11E-02	2.63E+03	0
20	MEAN	0	1.21E-02	2.10E+01	1.46E-06	3.43E-03	7.69E+03	0
	STD	0	1.30E-02	1.16E+01	3.20E-07	2.63E-03	1.07E+03	0
21	MEAN	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.79E-03	- 3.79E-03	2.01E-01	- 3.29E-03
	STD	1.36E-05	5.71E-18	4.45E-07	4.57E-19	4.57E-19	2.23E-01	1.11E-03
22	MEAN	9.98E-01	9.98E-01	1.30E+01	1.27E+01	1.55E+01	1.27E+05	7.03
	STD	1.13E-03	0	5.35E+01	1.87E-15	0	0	4.61
23	MEAN	3.98E-01	3.98E-01	1.84	3.98E-01	3.98E-01	3.98E-01	4.64E-01
	STD	1.58E-05	0	1.11	5.85E-17	1.00E-09	3.27E-04	7.80E-02
24	MEAN	3.73E-04	1.05E-03	1.23	2.91E-02	1.32E-03	3.01E-03	8.97E-03
	STD	9.98E-05	1.08E-04	1.51	3.17E-02	3.85E-04	3.29E-03	1.02E-02

Table 7 (continued)

f(x)		h-BOASOS	ABC	CS	FA	GA	MBO	MFO
25	MEAN	0	1.03E-28	2.06E+09	1.40E+09	1.00E-03	2.07E+08	0
	STD	0	1.52E-28	5.37E+08	2.54E+08	5.93E-04	2.19E+08	0
26	MEAN	0	4.73E-31	1.00E+10	8.46E-263	3.28E-05	2.95E+01	0
	STD	0	7.26E-31	0	0	3.93E-05	3.63E+01	0
27	MEAN	0	1.58E-12	7.84E-01	1.35E-31	2.27E-10	6.24E-16	5.00E-01
	STD	0	1.20E-12	1.97E-01	0	5.14E-10	5.94E-16	3.69E-05

Table 8 Simulation results of functions 28 to 35 functions of h-BOASOS and some other algorithms with NP=50 and 30 execution each with 10,000 iterations

f(x)		h-BOASOS	ABC	CS	FA	GA	MBO	MFO
28	MEAN	0	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.0009	5.00E-01
	STD	0	0	1.30E-04	2.95E-05	1.43E-05	0	3.69E-05
29	MEAN	0	1.72E-10	1.19E+02	0	4.21E-02	3.16E-16	0
	STD	0	1.23E-10	7.05E+01	0	1.17E-01	4.73E-16	0
30	MEAN	0	1.45E-08	1.40E+02	0	2.79E-09	1.73E-13	0
	STD	0	1.64E-08	1.56E+02	0	2.89E-09	2.57E-13	0
31	MEAN	0	4.93E+73	7.98E+05	1.67E+12	1.59E+02	3.59E+81	2.29E+03
	STD	0	9.88E+73	8.59E+05	2.28E+11	2.16E+02	1.05E+82	9.47E+02
32	MEAN	0	2.23E-11	5.25	0	1.17E-09	5.48E-04	7.24E-01
	STD	0	1.75E-11	4.52	0	1.18E-09	1.72E-01	1.02
33	MEAN	0	6.67E-01	1.65E+06	1.37E+03	1.11	1.06e+06	9.98E-01
	STD	0	0	3.56E+05	6.27E+02	1.37	7.12e+05	3.29E-03
34	MEAN	0	9.98E-01	4.74E+01	2.68	6.73	0.998	5.02
	STD	0	0	2.04E+01	1.92	4.74	0	3.44
35	MEAN	0	9.75E+07	1.00E+10	1.37E+12	3.42	3.80E+11	0
	STD	0	9.39E+07	0	2.82E+11	3.16	1.31E+11	0

Table 9 Comparison of experimental results of Tables 5 and 6, where the numbers represent the number of functions

Algorithms	Superior to to	Similar to to	Inferior to to
BOA	17	16	2
SOS	16	12	7
DE	23	5	7
PSO	29	3	3
JAYA	24	3	6
SCA	33	1	1

Table 10 Comparison of experimental results of Tables 7 and 8, where the numbers represent the number of functions

Algorithms	Superior to to	Similar to to	Inferior to to
ABC	28	2	5
CS	33	1	1
FA	22	7	6
GA	28	2	5
MBO	29	3	3
MFO	15	19	1

but the ranges of variables are considered as constraints. The schematic diagram of the system is shown in Fig. 7 The gear ratio is calculated as follows:

$$Gear\ ratio = \frac{(angular\ velocity\ of\ output\ shaft)}{(angular\ velocity\ of\ input\ shaft)}$$

The mathematical formulation of this problem is as follows:

$$Minimize\ f(x) = \left(\frac{1}{6.931} - \frac{x_c x_b}{x_a x_d} \right)^2$$

Subject to

$$12 \leq x_a, x_b, x_c, x_d \leq 60$$

This problem is solved with h-BOASOS, and the results are compared to MFO, ABC, MBA, GA, CS, and ISA in Table 14. Table 14 shows that the h-BOASOS algorithm finds the better optimal gear ratio value than MFO, ABC, MBA, GA CS, and ISA. The obtained result proves that h-BOASOS can be useful in solving discrete problems as well. All the results except h-BOASOS have been taken from [55].

Table 11 Friedman’s rank test with compared algorithms from Tables 5 and 6

Algorithms	h-BOASOS	BOA	SOS	DE	PSO	JAYA	SCA
Mean Rank	2.29	3.46	2.94	3.90	5.53	4.50	5.38
Rank	1	3	2	4	7	5	6

Table 12 Friedman’s rank test with compared algorithms from Tables 7 and 8

Algorithm	h-BOASOS	ABC	CS	FA	GA	MBO	MFO
Mean Rank	2.19	2.31	4.50	6.06	3.75	3.97	5.22
Rank	1	2	5	7	3	4	6

Table 13 Friedman’s rank test with all of the compared algorithms

Algorithm	h-BOASOS	BOA	SOS	DE	PSO	JAYA	SCA	ABC	CS	FA	GA	MBO	MFO
Mean Rank	3.49	5.16	4.40	5.87	9.12	6.57	8.51	3.78	8.07	11.25	7.26	7.69	9.82
Rank	1	4	3	5	11	6	10	2	9	13	7	8	12

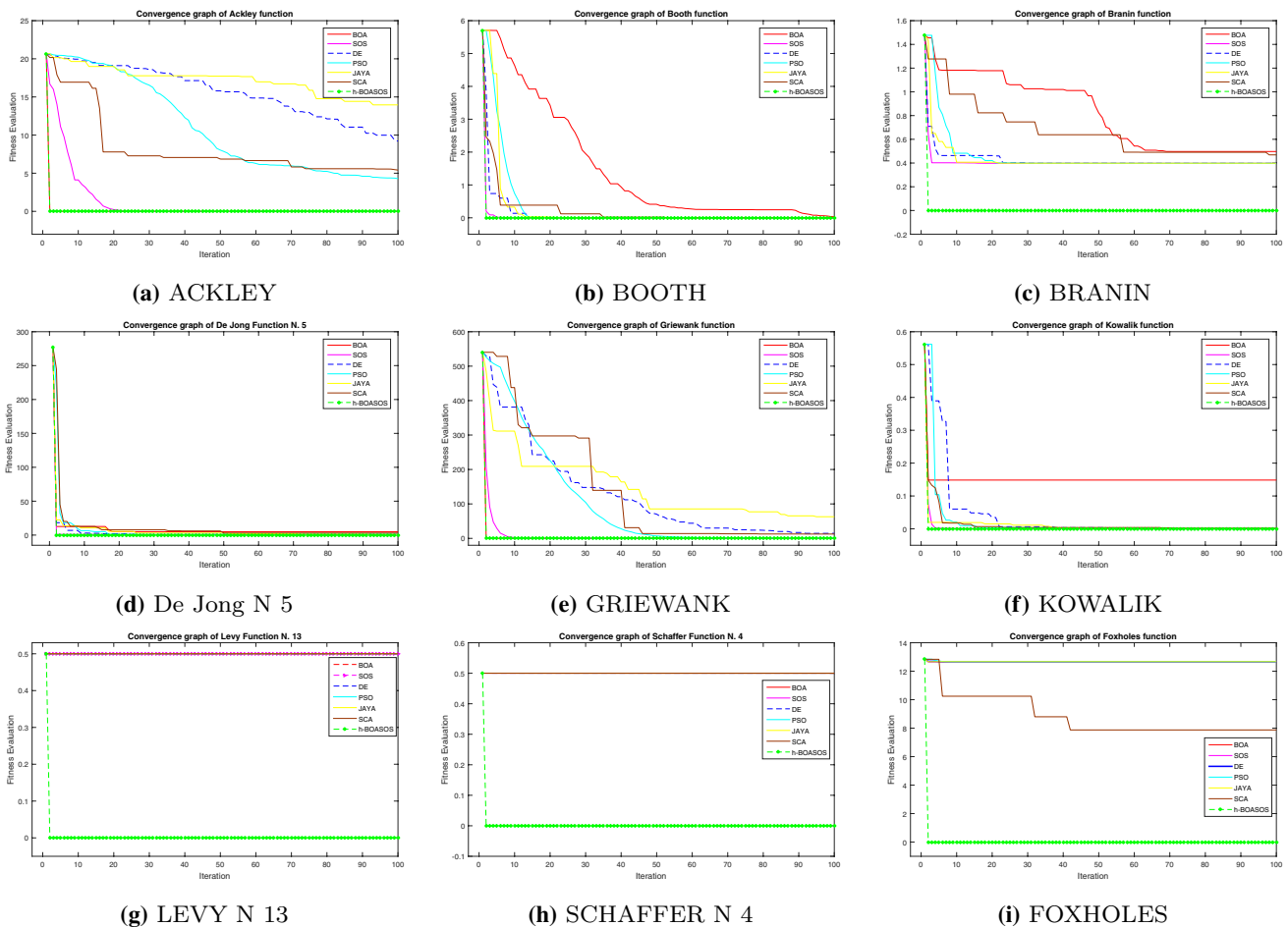


Fig. 2 Convergence graphs of h-BOASOS compared with BOA, SOS, DE, PSO, JAYA, and SCA

7.2 Cantilever beam design problem

A beam is a structural element that is capable of withstanding load primarily by resisting bending. Here the cantilever

beam has five square-shaped hollow elements. Figure 8 depicts that each element acts as one variable with the thickness as constant, so there are five structural parameters in total. It may be seen from Fig. 8 that a vertical load also acts

Table 14 Comparison of results on the gear train design problem

	h-BOASOS	MFO	ABC	MBA	GA	CS	ISA
x_a	48	43	49	43	49	43	33
x_b	27	19	16	169	16	16	15
x_c	16	16	19	19	19	19	13
x_d	44	49	43	49	43	49	41
$f(x)$	1.37e-19	2.70e-12	2.70e-12	2.70e-12	2.70e-12	2.70e-12	2.15e-08

to the free end of the beam (node 6), and the right side of the beam (node 1) is rigidly supported. Minimizing the weight of the beam is the objective here. One vertical displacement constraint is there that should be taken care of by the final optimal design. The mathematical formulation of the problem may be represented as:

$$\text{Minimize } f(x) = 0.06224(x_1 + x_2 + x_3 + x_4 + x_5);$$

$$\text{Subject to : } g(x) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 \leq 1;$$

Where $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$;

The problem has been solved by h-BOASOS, and the results are compared with some other methods viz., ALO, CS, SOS, CGA-I, CSA-II, and MMA. The results other than h-BOASOS are taken from [57]. It has been found from Table 15 that the proposed method shows better performance for this problem.

7.3 Car side impact design problem

The objective of this problem is to minimize the car’s total weight using eleven mixed variables while maintaining safety performance according to the standard. These variables represent the thickness and material of critical parts of the car. The 8th and the 9th variables are discrete, and these are material design variables, while the rest of the variables are continuous and represents thickness design variables.

The symbols $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}$ are used here to represent the variables thickness of B-pillar inner, the thickness of B-pillar reinforcement, the thickness of floor side inner, the thickness of cross members, the thickness of door beam, the thickness of door beltline reinforcement, the thickness of roof rail, the material of B-pillar inner, the material of floor side inner, barrier height, barrier hitting position respectively. The problem is subjected to ten inequality constraints. The car side impact design is considered a real case of a mechanical optimization problem with mixed discrete and continuous design variables. This problem can be mathematically described as:

$$a = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11})$$

$$\text{Min } f(a) = 1.98 + 4.90a_1 + 6.67a_2 + 6.98a_3 + 4.01a_4 + 1.78a_5 + 2.73a_7$$

Subject to

$$h_1(a) = 1.16 - 0.3717a_2a_4 - 0.00931a_2a_{10} - 0.484a_3a_9 + 0.01343a_6a_{10} \leq 1$$

$$h_2(a) = 0.261 - 0.0159a_1a_2 - 0.188a_1a_8 - 0.019a_2a_7 + 0.0144a_3a_5 + 0.0008757a_5a_{10} + 0.080405a_6a_9 + 0.00139a_8a_{11} + 0.00001575a_{10}a_{11} \leq 0.32$$

$$h_3(a) = 0.214 + 0.00817a_5 - 0.131a_1a_8 - 0.0704a_1a_9 + 0.03099a_2a_6 - 0.018a_2a_7 + 0.0208a_3a_8 + 0.121a_3a_9 - 0.00364a_5a_6 + 0.0007715a_5a_{10} - 0.0005354a_6a_{10} + 0.00121a_8a_{11} \leq 0.32$$

$$h_4(a) = 0.074 - 0.061a_2 - 0.163a_3a_8 + 0.001232a_3a_{10} - 0.166a_7a_9 + 0.227a_2^2 \leq 0.32$$

$$h_5(a) = 28.98 + 3.818a_3 - 4.2a_1a_2 + 0.0207a_5a_{10} + 6.63a_6a_9 - 7.7a_7a_8 + 0.32a_9a_{10} \leq 32$$

$$h_6(a) = 33.86 + 2.95a_3 + 0.1792a_{10} - 5.05a_1a_2 - 11.0a_2a_8 - 0.0215a_5a_{10} - 9.98a_7a_8 + 22.0a_8a_9 \leq 32$$

$$h_7(a) = 46.36 - 9.9a_2 - 12.9a_1a_8 + 0.1107a_3a_{10} \leq 32$$

$$h_8(a) = 4.72 - 0.5a_4 - 0.19a_2a_3 - 0.0122a_4a_{10} + 0.009325a_6a_{10} + 0.000191a_{11}^2 \leq 4$$

$$h_9(a) = 10.58 - 0.674a_1a_2 - 1.95a_2a_8 + 0.02054a_3a_{10} - 0.0198a_4a_{10} + 0.028a_6a_{10} \leq 9.9$$

$$h_{10}(a) = 16.45 - 0.489a(3)a_7 - 0.843a_5a_6 + 0.0432a_9a_{10} - 0.0556a_9a_{11} - 0.000786a_{11}^2 \leq 15.7$$

where, $0.5 \leq a(1) - a(7) \leq 1.5$
 $a(8), a(9) \in (0.192, 0.345)$
 $- 30 \leq a(10), a(11) \leq 30$

Table 15 Comparison of results on the cantilever beam design problem

	h-BOASOS	ALO	SOS	CS	MMA	GCA-I	GCA-II
x_1	1.298659	6.01812	6.01878	6.0089	6.0100	6.0100	6.0100
x_2	15.3227301	5.31142	5.30344	5.3049	5.3000	5.30400	5.3000
x_3	57.345186	4.48836	4.49587	4.5023	4.4900	4.4900	4.4900
x_4	23.2403325	3.49751	3.49896	3.5077	3.4900	3.4980	3.4900
x_5	87.429983	2.158329	2.15564	2.1504	2.1500	2.1500	2.1500
$f(x)$	1.298659	1.33995	1.33996	1.33999	1.3400	1.3400	1.3400

Table 16 Comparison of results on the car-side impact design problem

Method	h-BOASOS	WOAmM	ABC	PSO	MFO	ALO	ER-WCA
a_1	0.81	0.50	0.50	0.50	0.50	0.50	0.50
a_2	0.75	0.9601	1.0624	1.1165	1.1165	1.1159	1.1186
a_3	0.54	0.8674	0.5148	0.50	0.50	0.50	0.50
a_4	0.82	0.50	1.4491	1.3018	1.3019	1.3028	1.2984
a_5	0.63	0.6433	0.50	0.50	0.50	0.50	0.50
a_6	0.82	1.8295	1.50	1.50	1.50	1.50	1.50
a_7	0.52	0.50	0.50	0.50	0.50	0.50	0.50
a_8	0.19	0.3450	0.3450	0.3450	0.3450	0.3450	0.3450
a_9	0.17	0.3450	0.192	0.3450	0.3450	0.192	0.192
a_{10}	- 11.65	-30	-29.34	19.52	-19.53	19.63	-19.14
a_{11}	- 4.75	-4.8066	0.74109	-0.019	0.000006	0.02364	-0.01527
$f(a)$	20.5422	21.4034	23.175	22.842	22.842	22.842	22.842

To judge the efficiency of the newly proposed h-BOASOS in the car side impact design problem, the results have been compared to that of 11 other state-of-the-art algorithms viz., WOAmM, ABC, PSO, MFO, ALO, ER-WCA, GWO, WCA, MBA, SSA, and WOA. Tables 15, 16 and 17 shows the result found by h-BOASOS and comparison with the results found in literature [55]. Results are taken using a population size of 35, and 850 is the maximum number of iterations in one run, whereas the proposed h-BOASOS find the minimum solution in 50 iterations. $f(x)$ here represents the value of the optimal weight of the car.

7.4 Three bar truss design problem

The purpose of the problem is to reduce the weight of the bar structures. Stress, deflection, and buckling constraints are the limitations of the problem. The objective function of the problem is a non-linear function with three non-linear limitations. The mathematical formulation of the problem is given below:

$$\text{Minimize } f(x) = Lx_2 + 2\sqrt{2}x_1$$

where

Table 17 Comparison of results on car-side impact design problem

Method	h-BOASOS	GWO	WCA	MBA	SSA	WOA
a_1	10.81	0.50	0.50	0.50	0.50	0.50
a_2	0.75	1.1114	1.1155	1.1172	1.1093	1.1080
a_3	0.54	0.50	0.50 0.50	0.50	0.5344	
a_4	0.82	1.3122	1.3034	1.3008	1.3148	1.3057
a_5	0.63	0.5012	0.50	0.50	0.50	0.50
a_6	0.82	1.50	1.50	1.4999	1.4999	1.4738
a_7	0.52	0.50	0.50	0.50	0.50	0.50
a_8	0.19	0.3450	0.3450	0.3450	0.3450	0.3450
a_9	0.17	0.192	0.192	0.3450	0.192	0.192
a_{10}	- 11.65	-20.60	-19.69	-19.40	-20.82	-19.69
a_{11}	- 4.75	-0.25531	-0.2385	-0.3732	0.44129	3.48169
$f(a)$	20.5422	22.852	22.843	22.846	23.042	23.042

$$h_1(x) = \frac{x_2}{(2x_2x_1 + \sqrt{2}x_1^2)}P - \rho \leq 0$$

$$h_2(x) = \frac{x_2 + \sqrt{2}x_1}{(2x_2x_1 + \sqrt{2}x_1^2)}P - \rho \leq 0$$

$$h_3(x) = \frac{1}{(x_1 + \sqrt{2}x_2)}P - \rho \leq 0$$

where, $0 \leq x_1, x_2 \leq 1$ and $P = 2, L = 100, \rho = 2$.

To evaluate the performance of the h-BOASOS on this problem, the result evaluated by the proposed method is compared to ten other algorithms, among which few are collected from the literature [50]. Table 18 contains the best value found by the h-BOASOS and other algorithms. Here to check the convergence speed of the proposed method total iteration has been set to 35 with 1000 iteration. It seems that the proposed method finds a new optimal solution, ‘150.451’, with a new set of parameters value, [0.41471 0.33153] to this problem.

8 Cantilever retaining wall

In the optimum design of the structure, three stages are generally considered: structural modeling, optimum design modeling, and the optimization algorithm. In the first stage, i.e., structural modeling, the formulation of the problem is done, and the objective function of the problem is defined without violating the constraints. The problem considered

for optimization is solved in the past by using optimtool [58]. Figure 3 shows a sketch of the cantilever retaining wall model showing variables used for optimization.

8.1 Structural modeling

In most of the structural problems, the objective function is defined based on the weight and cost of the structure depending on some constraints. Weight is calculated by assuming a density of concrete and steel, and the cost is estimated as the combined cost of reinforcement and concrete. The optimization equation can be formulated as:

Minimize $f(x)$

Subject to

$$g_i(x) = 0 \dots \dots i = 1, 2, \dots, p$$

$$h_j(x) \leq 0 \dots \dots j = 1, 2, \dots, m$$

$$L_k(x) \leq X_k(x) \leq U_k(x) \dots k = 1, 2, \dots, n$$

where, objective function is given by $f(X)$, $g_i(X)$ and $h_j(X)$ are equality and inequality constraints respectively, L_k and U_k are lower and upper bounds of the k th variable.

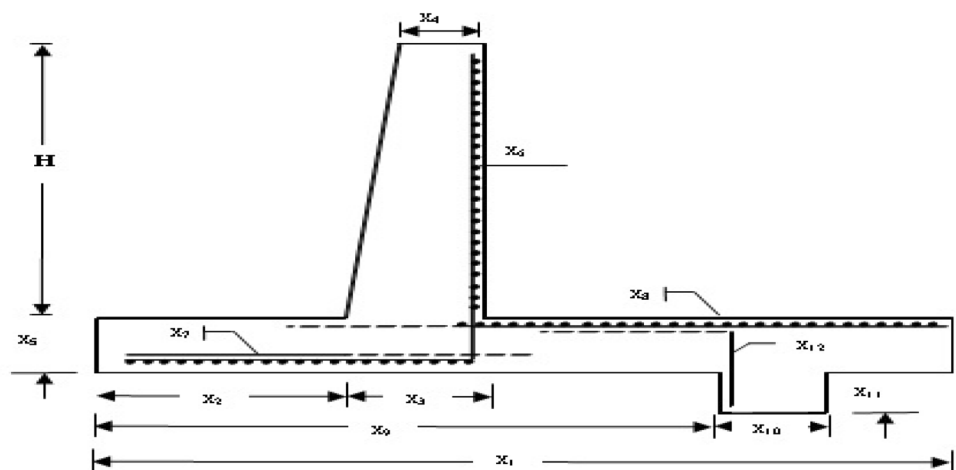
8.2 Optimum design modeling

In this stage, the parameters of the problems are studied in-depth so that parameters, design variables, constraints, and the objective function can be decided.

Table 18 Comparison of results on three bar truss design problem

Method	h-BOASOS	MFO	DEDS	PSO-DE	MBA	Ray and Sain	Tsa	CS
a_1	10.41471	0.78824	0.78868	0.78868	0.78857	0.795	0.788	0.78867
a_2	0.33153	0.40947	0.40825	0.40825	0.40856	0.395	0.408	0.40902
$f(a)$	150.45075	263.89598	263.89584	263.89584	263.89585	264.3	263.68	263.9716

Fig. 3 Retaining wall model representing variables used for optimization



8.2.1 Design variables

Figure 3 shows design variables for retaining wall, include: width of the base of retaining wall (X_1), toe projection (X_2), thickness at the bottom of the stem (X_3), thickness at the top of the stem (X_4), thickness of base slab (X_5), area of vertical reinforcement in the stem (X_6), area of horizontal reinforcement in the toe (X_7), area of horizontal reinforcement in the heel (X_8), distance from toe to the front of the shear key (X_9), shear key width (X_{10}), shear key depth (X_{11}), area of vertical reinforcement in the base shear key (X_{12}).

8.2.2 Constraints

Retaining structures seeks designs that provide safety and stability against different types of failure modes and meet the terms with concrete building code requirements. These requirements can be categorized into four general groups of design constraints: capacity, stability, reinforcement configuration, and geometric limitations. Design constraints are imposed as penalties on the objective functions and are non-zero only when violated. If the design is reasonable, the sum of the constraint penalties will be zero. The design constraints are based on geotechnical and structural requirements as per IS code 456:2000. There are different types of failure modes such as bearing capacity, sliding, and overturning, and for these modes of failure minimum factor of safety should be provided by retaining wall (Table 20).

i. Overturning Failure Mode:

$$g_1(x) = F_0 - \left(\sum M_{vo} / \sum M_{ho} \right) \leq 0$$

where, F_0 is Factor of safety against overturning, $\sum M_{vo}$ represents the total moment of vertical forces that resist overturning about toe, $\sum M_{ho}$ is the total horizontal moment of forces that tends to overturn about the toe.

ii. Bearing Capacity:

$$g_2(x) = P_{min} \leq 0$$

$$g_3(x) = P_{max} - SBC \leq 0$$

in which, P_{min} and P_{max} is minimum, and maximum contact pressure at the interface between the foundation soil and the wall structure and safe bearing capacity of the soil is represented as SBC.

iii. Sliding Failure Mode

$$g_4(x) = F_s - \left[\left(\sum V * \mu + H_p \right) / \sum H \right] \leq 0$$

where, F_s represents factor of safety against sliding, $\sum H$ and $\sum V$ are total horizontal and vertical driving forces respectively and H_p denotes horizontal passive pressure.

iv. Tension Failure Mode

$$g_5(x) = E - (B/6) \leq 0$$

in which, eccentricity of the resultant force is given by E and base width of retaining wall by B.

v. Moment failure mode

The flexural strength M_{rs} is computed as:

$$M_{rs} = 0.87A_s f_y (d_s - 0.416X_u)$$

$$X_u = 0.87A_s f_y (0.36f_{ck}b)$$

$$g_6(x) = M_s - M_{rs} \leq 0$$

where, X_u is the location of the neutral axis, A_s is c/s area of steel, d_s is effective depth of stem, f_y is the yield strength of reinforcement, M_s represents the maximum bending moment at the face of the wall, M_{rs} is flexural strength of stem.

$$g_7(x) = M_t - M_{rt} \leq 0$$

$$g_8(x) = M_h - M_{rh} \leq 0$$

$$g_9(x) = M_k - M_{rk} \leq 0$$

where, M_t and M_h are maximum bending moment at the junction of the stem with toe slab and heel slab, Maximum bending moment at base shear key, M_{rt} and M_{rh} are Modulus of rupture of the toe slab, heel slab, M_k is the moment at the base of the shear key.

vi. Shear failure mode

$$g_{10}(x) = V_s - V_{us} \leq 0$$

$$g_{11}(x) = V_t - V_{ut} \leq 0$$

$$g_{12}(x) = V_h - V_{uh} \leq 0$$

$$g_{13}(x) = V_k - V_{uk} \leq 0$$

where, V_s , V_t , V_h , and V_k are the maximum shear carrying capacity of the stem, toe, heel, and shear key, respectively. Similarly, V_{us} , V_{ut} , V_{uh} and V_{uk} are the shear capacity of the concrete stem, toe, heel, and shear key.

vii. Criteria for minimum reinforcement

$$g_{14}(x) = (0.12/100b) bD_s - A_s \leq 0$$

$$g_{15}(x) = (0.12/100b) bD_s - A_t \leq 0$$

$$g_{16}(x) = (0.12/100b) bD_s - A_h \leq 0$$

where, the base width of the retaining wall is given by b, D_s and D are thickness at the bottom of stem and base slab, A_s , A_t , and A_h are the area of reinforcement in the stem, toe, and heel.

viii. Criteria for maximum reinforcement

$$g_{17}(x) = A_s - (4/100) bD_s \leq 0$$

$$g_{18}(x) = A_t - (4/100) bD \leq 0$$

$$g_{19}(x) = A_h - (4/100) bD \leq 0$$

ix. Additional geometric criteria

In order to prevent infeasible retaining wall dimensions, some additional geometrical constraints have been added.

$$g_{20}(x) = X_2 + X_3 - X_1 \leq 0$$

$$g_{21}(x) = X_9 + X_{10} - X_1 \leq 0$$

x. Lower and upper bound constraints

The derived constraint expressions are found to be highly nonlinear in the design variables.

8.2.3 Objective function

As mentioned earlier, the objective of the problem is the minimization of weight and cost. The forms of two objective functions for this sort of optimization problem in structural engineering are reliable. The cost function $f(cost)$ is given by:

$$f_{cost} = C_s W_s + C_c V_c$$

where C_s is the unit cost of steel and concrete, W_s is the weight of reinforcement per unit length of the wall, and V_c is the volume of concrete per unit length. The second objective function depends exclusively on the weight of the materials. The weight function $f(weight)$ is:

$$f_{weight} = W_s + 100V_c\gamma_c$$

Where, γ_c is the unit weight of concrete, and a factor of 100 is used for consistency of units

9 Results and Discussions

In this section, the results of minimum cost and minimum weight for cantilever retaining wall of height 3.2 m and 6.3 m are obtained by the newly proposed h-BOASOS algorithm, and some discussions on the obtained results are

performed. Table 21 shows different input parameters for the cantilever retaining wall.

9.1 For 3.2 m and 6.3 m high retaining wall

For the retaining wall with a height of 3.2 m, lower and upper bounds of design variables for the two objective functions which are constructed to obtain cost and weight are presented in Table 19.

Values of design constraints for a 3.2 m high retaining wall are presented in Table 23. Again, Table 22 depicts the optimized weight and cost for the entire structure. Results obtained from the h-BOASOS algorithm for the optimized weight of steel and volume of concrete required for

Table 20 Constraints of cantilever retaining wall

Inequality	Failure mode constraints
$g_1(X)$	Overtuning failure mode
$g_2(X)$	Bearing failure mode (min)
$g_3(X)$	Bearing failure mode (max)
$g_4(X)$	Sliding failure mode
$g_5(X)$	Tension failure mode
$g_6(X)$	Moment failure mode at bottom of stem
$g_7(X)$	Moment failure mode at toe
$g_8(X)$	Moment failure mode at heel
$g_9(X)$	Moment at shear key
$g_{10}(X)$	Shear at bottom of stem
$g_{11}(X)$	Shear at Toe
$g_{12}(X)$	Shear at heel
$g_{13}(X)$	Shear at shear key
$g_{14}(X)$	Criteria for minimum reinforcement
$g_{15}(X)$	Criteria for maximum reinforcement
$g_{16}(X)$	Additional geometric constraints

Table 19 Lower and upper bounds of design variables

Design variables	Lower bounds	Upper bounds
Width of the base X_1	$X_1 = 0.4 * h * (12/11)$	$X_1 = (0.7 * h)/0.9$
Projection of toe X_2	$X_2 = [0.4 * h * (12/11)]/3$	$X_2 = [(0.7 * h)/0.9]/3$
Thickness at the bottom of the stem X_3	$X_3 = 0.2$	$X_3 = (h/0.9)/10$
Thickness at the top of the stem X_4	$X_4 = 0.2$	$X_4 = 0.2$
Thickness of base slab X_5	$X_5 = [h * (12/11)]/12$	$X_5 = (h/0.9)/10$
Area of vertical reinforcement in the stem, per unit length of wall X_6	$X_6 = 0.0012 * X_3$	$X_6 = 0.04 * X_3$
Area of horizontal reinforcement in the toe X_7	$X_7 = 0.0012 * X_5$	$X_7 = 0.04 * X_5$
area of horizontal reinforcement in the heel of wall X_8	$X_8 = 0.0012 * X_5$	$X_8 = 0.04 * X_5$
The distance from toe to the front of the base shear key X_9	$X_9 = 0.4 * h * (12/11)$	$X_9 = (0.7 * h)/0.9$
Shear key width X_{10}	$X_{10} = 0.3$	$X_{10} = (h/0.9)/10$
Depth of the shear key X_{11}	$X_{11} = 0.3$	$X_{11} = (h/0.9)/10$
area of vertical reinforcement in the base shear key X_{12}	$X_{12} = 0.0012 * X_{10}$	$X_{12} = 0.04 * X_3$

Table 21 Input parameters for the retaining wall problem

Input Parameter	Unit	Symbol	Value
Stem Height	<i>m</i>	H	3.2, 6.3
Angle of internal friction	<i>Degree</i>	ϕ	35
Surcharge load	<i>kN/m²</i>	Q	10
Backfill slope	<i>Degree</i>	β	0
Yield strength of steel	<i>kN/m²</i>	f_y	$500 * 10^3$
Characteristic strength of concrete	<i>kN/m²</i>	f_{ck}	$25 * 10^3$
Density of soil	<i>kN/m³</i>	ρ_{soil}	17
Unit weight of concrete	<i>kN/m</i>	Γ_D	25
Concrete cover	<i>m</i>	cover	0.025
Safe bearing capacity of soil	<i>kN/m</i>	SBC	250
Coefficient of friction under base	NA	μ	0.55
Factor of safety against overturning	NA	F_o	1.4
Factor of safety against sliding	NA	F_s	1.4
Factor of safety for bearing capacity	NA	F_b	3
Cost of steel	<i>Rs/kg</i>	C_s	60
Cost of concrete	<i>Rs/m</i>	C_c	80
% minimum steel	%	P_{min}	0.12
% maximum	%	P_{max}	4

construction to be safe have been shown in Table 24. The minimum weight obtained using the h-BOASOS algorithm is 2630.8284 Kg/m, and the minimum cost is 1895.3257 Rs/m (1 \$ = Rs.74.2 approx).

In Table 25, optimized results of design variables are presented for 6.3 m high retaining wall.

Values of design constraints for 6.3 m high retaining wall are presented in Tables 26 and 27 shows optimized weight and cost for the entire structure. Minimum weight obtained by using h-BOASOS algorithm is 9093.9172 Kg/m and minimum cost is Rs.3359.5348/m.

Figure 2 shows the optimum value of cost function and weight function at different heights of cantilever retaining wall from 3 m to 6 m obtained from the h-BOASOS Algorithm.

The range of the stem height (3–6 m) shows that for higher values of height, the optimum cost and weight become more sensitive to variations in the stem height. This is more apparent for the cost function. Higher stem height produces higher values of the objective function. It is obvious that weight and cost will increase as the retaining wall’s height increases, which is reflected in Fig. 2. The optimum values for both objective models are shown for different heights from 3 to 6 m; the optimum weight increases 3.42 times while the optimum cost by 1.53 times.

Figure 4 shows the optimum value of concrete required for both the objective functions at different heights of the cantilever retaining wall from 3 to 6 m. Although it does not follow a particular trend of increase or decrease, a weight function optimum volume of concrete for 6 m high retaining wall is almost 1.79 times greater than that of 3 m high. Similarly, for cost function, the optimum volume of concrete for a 6 m high retaining wall is almost 3.43 times greater than a 3 m high wall.

Figure 5 represents the optimum value of the weight of steel obtained for both objective functions at different heights of the cantilever retaining wall from 3 to 6 m. For the weight function, the optimum weight of steel for a 6 m high retaining wall is almost 0.73 times lesser than 3 m high. Similarly, for cost function, the optimum volume of concrete for a 6 m high retaining wall is almost 0.377 times greater than that of 3 m high (Fig. 6).

Table 22 Optimum values of design variables for 3.2 m high retaining wall

Design variables	Unit	Lower bounds	Upper bounds	For optimized cost	For optimized weight
X_1	<i>m</i>	1.396	2.488	2.4471	2.0541
X_2	<i>m</i>	0.466	0.8296	0.6258	0.5735
X_3	<i>m</i>	0.2	0.356	0.3154	0.3435
X_4	<i>m</i>	0.2	0.2	0.2	0.2
X_5	<i>m</i>	0.291	0.356	0.3103	0.3484
X_6	<i>m</i>	0.00024	0.0142	0.0048	0.0003
X_7	<i>m</i>	0.000349	0.0142	0.0007	0.0004
X_8	<i>m</i>	0.000349	0.0142	0.0016	0.0004
X_9	<i>m</i>	1.3964	2.4889	2.0005	2.0447
X_{10}	<i>m</i>	0.3000	0.3556	0.3535	0.3
X_{11}	<i>m</i>	0.3000	0.3556	0.3293	0.3
X_{12}	<i>m</i>	0.00036	0.142	2.4471	0.0004

Table 23 Values of constraints at optimum values of design variables for 3.2 m high retaining wall

Constraint	For minimum cost	For minimum weight
$g_1(X)$	-4.9038	-11.4667
$g_2(X)$	-0.029	0.0194
$g_3(X)$	-0.6152	-0.8939
$g_4(X)$	-5.2007	-11.1403
$g_5(X)$	-0.5891	-0.6703
$g_6(X)$	121787.6	2652.872
$g_7(X)$	115450.7	3749.243
$g_8(X)$	117770.2	3758.298
$g_9(X)$	0	0
$g_{10}(X)$	-0.0374	-0.0374
$g_{11}(X)$	0.0002	-0.0002
$g_{12}(X)$	-0.0014	-0.0012
$g_{13}(X)$	0	0
$g_{14}(X)$	0.0006	0.0009
$g_{15}(X)$	0.0006	0.0009
$g_{16}(X)$	0.0006	0.0009
$g_{17}(X)$	-0.0091	-0.028
$g_{18}(X)$	-0.0092	-0.0283
$g_{19}(X)$	-0.009	-0.0283
$g_{20}(X)$	-0.5867	-1.1372
$g_{21}(X)$	0.7035	0.2907

Table 24 Minimum values of objective function for 3.2 m high retaining wall

Objective function	Unit	Optimum value	Weight of steel (kg/m)	Volume of concrete (m^3/m)
For minimum weight	Kg/m	2630.8284	22.1173	1.0434
For minimum cost	Rs/m	1895.3257	25.5198	1.2955

9.2 Comparison of results

In this section, the optimum values of design variables, the optimum volume of concrete, and optimum cost and weight of retaining wall obtained by h-BOASOS are compared with the optimitool, BOA, and SOS results.

Tables 19, 20, 21 and 22 present the comparison of results obtained from Optim tool [58], BOA, SOS and proposed methodology, i.e., h-BOASOS. Optimized geometrical parameters for which weight and cost are minimum, the optimum weight of steel, and the concrete volume are presented in Tables 19, 20, 21 and 22. From these tables, it is seen that the proposed algorithm h-BOASOS gives efficient results for both cost and weight function for 3.2 m and 6.3 m height of cantilever retaining wall. The optimum weight of the 3.2

Table 25 Optimum values of design variables for 6.3 m high retaining wall

Design variables	Unit	Lower Bounds	Upper Bounds	For Optimized Cost	For Optimized Weight
X_1	<i>m</i>	2.749	4.90	3.278	3.7016
X_2	<i>m</i>	0.916	1.633	1.3088	0.9285
X_3	<i>m</i>	0.2	0.7	0.6867	0.4636
X_4	<i>m</i>	0.2	0.2	0.2	0.2
X_5	<i>m</i>	0.573	0.70	0.6879	0.604
X_6	<i>m</i>	0.00024	0.028	0.0003	0.0244
X_7	<i>m</i>	0.000688	0.028	0.0007	0.0134
X_8	<i>m</i>	0.000688	0.028	0.0007	0.027
X_9	<i>m</i>	2.7490	4.9	2.9598	3.8216
X_{10}	<i>m</i>	0.3	0.7	0.3	0.3
X_{11}	<i>m</i>	0.3	0.7	0.3	0.3
X_{12}	<i>m</i>	0.00024	0.028	0.0004	0.0004

Table 26 Values of constraints at optimum values of design variables for 6.3 m high retaining wall

Constraint	For minimum cost	For minimum weight
$g_1(X)$	-5.1468	-10.0623
$g_2(X)$	-0.0056	0.2593
$g_3(X)$	-1.2683	-1.7422
$g_4(X)$	-5.7392	-11.3786
$g_5(X)$	-1.0975	-1.0741
$g_6(X)$	2838.869	260395.2
$g_7(X)$	7268.645	141971.4
$g_8(X)$	7298.247	287488.6
$g_9(X)$	0	0
$g_{10}(X)$	-0.145	-0.145
$g_{11}(X)$	-0.0004	-0.0021
$g_{12}(X)$	-0.0022	0.0001
$g_{13}(X)$	0	0
$g_{14}(X)$	0.0028	0.002
$g_{15}(X)$	0.0028	0.0027
$g_{16}(X)$	0.0028	0.0026
$g_{17}(X)$	-0.0898	-0.0443
$g_{18}(X)$	-0.0895	-0.0761
$g_{19}(X)$	-0.0895	-0.0625
$g_{20}(X)$	-1.2825	-2.3096
$g_{21}(X)$	-0.0182	0.4201

m retaining wall obtained from the h-BOASOS algorithm is 3115.34 Kg/m, and its Optimum Cost is Rs.11895.33/m. While, Optimum Weight of 6.3 m is 3115.34 Kg/m and it's Optimum Cost is 11895.33 Rs/m (1 \$ = Rs.74.2 approx) (Tables 28, 29, 30, 31).

Table 27 Minimum values of objective function for 6.3 m high retaining wall

Objective function	Unit	Optimum value	Weight of steel (kg/m)	Volume of concrete (m^3/m)
For minimum weight	kg/m	9093.9172	45.64164	3.6193
For minimum cost	Rs/m	3359.5348	73.3119	3.6495

10 Conclusion

In this envisage, the cantilever retaining wall problem is solved by a newly proposed hybrid algorithm h-BOASOS. The proposed algorithm is tested on a suite of thirty-five classical benchmark functions. The obtained results are then compared with a wide variety of popular optimization algorithms and found that the proposed method outperforms the compared algorithms in numerical results. To support our claim, we have performed the Friedman rank test and found that the rank of the proposed algorithm is the least. To examine the convergence speed of the h-BOASOS, some of the convergence graphs are plotted and found that the proposed algorithm converges faster than the compared algorithms. Further, two real-world engineering design problems obtained from literature are also solved by the suggested algorithm. The experimental results prove the efficiency of the proposed algorithm over several state-of-the-art

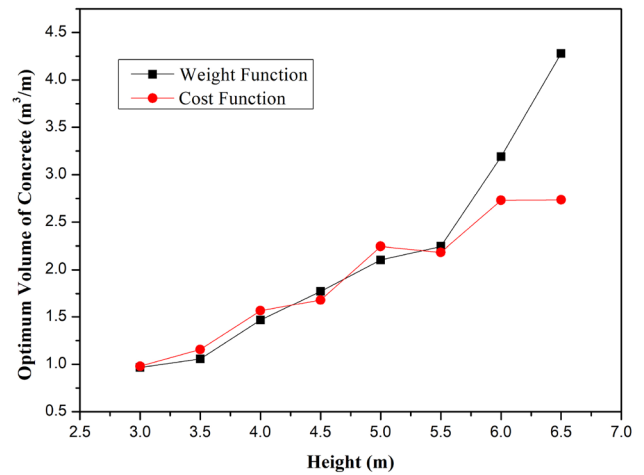
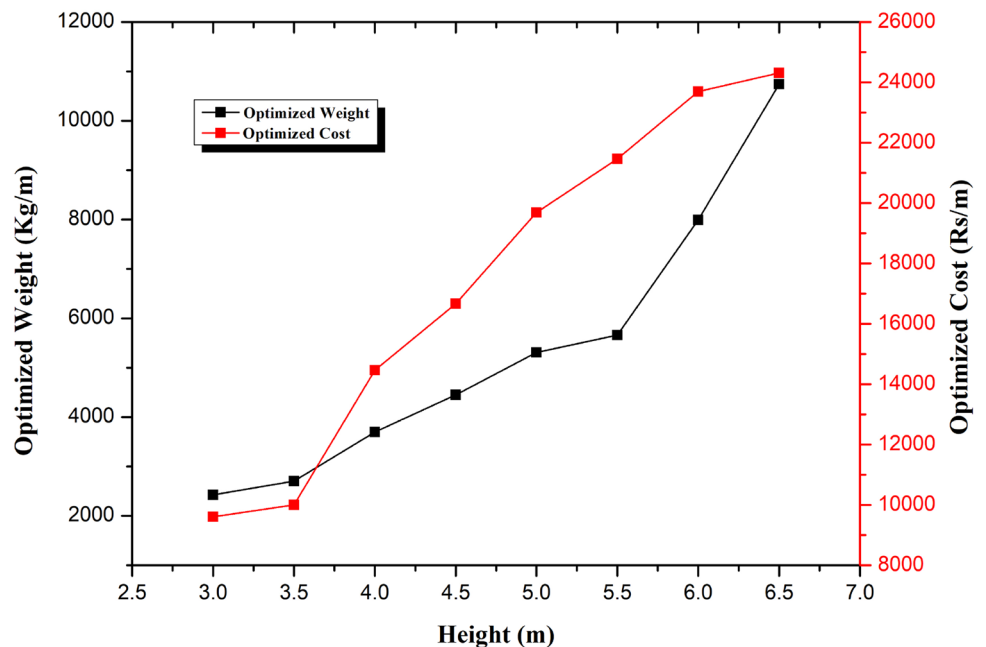


Fig. 5 Comparison of optimum volume of concrete for both the objective function at different stem height of cantilever retaining wall

algorithms. Finally, the reduction of weight and cost of retaining wall is achieved successfully by the newly proposed algorithm. Using h-BOASOS, for 3.2 m and 6.3 m high retaining wall, approximately 8.75% and 5.43% material weight is reduced compared to the results obtained from the optimtool, 19.6% and 2.05% compared to SOS. Similarly, in the case of cost obtained from h-BOASOS, the cost saved for the retaining wall of heights 3.2 m and 6.3 m are 14.02% and 56.53% compared to results obtained from the optimtool. While, as compared to SOS, cost saved is 14.7% and 2.2%, respectively.

Fig. 4 Comparison of cost function at different stem height of cantilever retaining wall



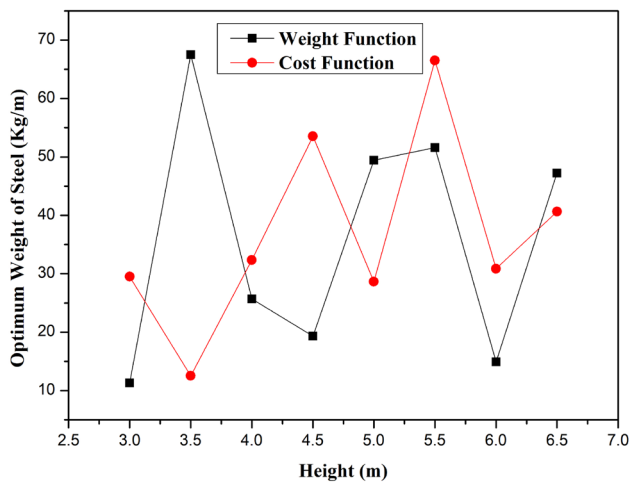


Fig. 6 Comparison of optimum weight of steel for both the objective function at different stem height of cantilever retaining wall

The researchers are always involved in exploring new hypotheses and simultaneously seeking to refine existing theories, so there is always space for finding new studies and improving existing research. Several further extensions can be made from the review of the research discussed in this article. Some of those are listed below:

- The present study may be upgraded by considering the adaptive or self-adaptive parameter setting of both the

component algorithms. Specifically, the switching probability of BOA and the beneficial factors of SOS can be taken adaptively.

- The concept of fuzzy and its extensions may be incorporated in the parameter setting. For example, fuzzy-adaptive, neighborhood-based, fitness distance-based, etc., may be combined with this study to improve its efficiency further.
- The mathematical foundation of the metaheuristic algorithms has not been widely studied so far. So, mathematical analysis of stability, convergence, divergence, complexity, etc., may be done for better understanding.
- Depending upon the physiognomies of the design variables (integer, mixed-integer, binary, discrete, etc.), some modified algorithms can be established to solve real-life industrial problems.
- The proposed algorithm may further be extended to the multimodal and multi and many-objective scenario.
- The suggested algorithm may be applied to the optimization of complex systems such as designing the structures of civil engineering, business models, inventory control, transportation problems, power system problem, simulation and planning, controlling, and scheduling problems, etc.

Table 28 Comparison of results—optimum weight as objective function for 3.2 m high retaining wall

	Unit	Optimtool	BOA	SOS	h-BOASOS
X_1	<i>m</i>	1.7594	1.9223	1.4535	2.0541
X_2	<i>m</i>	0.5979	0.6432	0.6603	0.5735
X_3	<i>m</i>	0.301	0.2513	0.2153	0.3435
X_4	<i>m</i>	0.2	0.2	0.2	0.2
X_5	<i>m</i>	0.291	0.3103	0.3439	0.3484
X_6	m^2	0.000519	0.0003	0.0099	0.0003
X_7	m^2	0.000349	0.0004	0.0079	0.0004
X_8	m^2	0.000349	0.0004	0.0042	0.0004
X_9	<i>m</i>		1.9175	1.8283	2.0447
X_{10}	<i>m</i>		0.3	0.3	0.3
X_{11}	<i>m</i>		0.3	0.3	0.3
X_{12}	<i>m</i>		0.0004	0.0004	0.0004
Volume of concrete	m^3/m	1.1212	1.1543	1.2943	1.2374
Weight of steel	<i>kg/m</i>	80.1159	5.01467	36.7628	21.7159
Min.weight	<i>kg/m</i>	2883.2	4.0303e+08	3272.6549	2630.8284

Table 29 Comparison of results—optimum cost as objective function for 3.2 m high retaining wall

	Unit	Optimtool	BOA	SOS	h-BOASOS
X_1	<i>m</i>	1.7594	2.3566	2.1745	1.6842
X_2	<i>m</i>	0.5979	0.7559	0.6057	0.7945
X_3	<i>m</i>	0.301	0.2638	0.2622	0.3031
X_4	<i>m</i>	0.2	0.2	0.2	0.2
X_5	<i>m</i>	0.291	0.3084	0.3376	0.295
X_6	m^2	0.000519	0.0003	0.013	0.0114
X_7	m^2	0.000349	0.0004	0.0006	0.0108
X_8	m^2	0.000349	0.0004	0.001	0.011
X_9	<i>m</i>		1.6735	1.6703	2.0876
X_{10}	<i>m</i>		0.3	0.3	0.3
X_{11}	<i>m</i>		0.3	0.3	0.3
X_{12}	<i>m</i>		0.0004	0.0004	0.0004
Volume of concrete	m^3/m	1.3136	1.2645	1.4361	1.2955
Weight of steel	<i>kg/m</i>	51.4396	4.9471	35.9331	25.5198
Min. cost	<i>Rs/m</i>	13585	3.8858e+08	13644.9537	11895.3257

Table 30 Comparison of results—optimum weight as objective function for 6.3 m high retaining wall

	Unit	Optimtool	BOA	SOS	h-BOASOS
X_1	<i>mm</i>	3.3108	2.8645	3.3289	3.7016
X_2	<i>m</i>	1.1057	1.0459	1.3474	0.9285
X_3	<i>m</i>	0.3060	0.2427	0.5881	0.4636
X_4	<i>m</i>	0.2	0.2	0.2	0.2
X_5	<i>m</i>	0.5910	0.6176	0.6472	0.604
X_6	m^2	0.004639	0.0004	0.0006	0.0244
X_7	m^2	0.000702	0.0008	0.001	0.0134
X_8	m^2	0.001252	0.0008	0.0009	0.027
X_9	<i>m</i>		3.0313	3.5485	3.8216
X_{10}	<i>m</i>		0.3	0.3	0.3
X_{11}	<i>m</i>		0.3	0.3	0.3
X_{12}	<i>m</i>		0.0004	0.0004	0.0004
Volume of concrete	m^3/m	3.5506	2.8945	3.6942	3.6193
Weight of steel	<i>kg/m</i>	739.390	8.82176	49.2574	45.64164
Min.weight	<i>kg/m</i>	9615.8	1.3174e+09	9284.9191	9093.9172

Table 31 Comparison of results—optimum weight as objective function for 6.3 m high retaining wall

	Unit	Optimtool	BOA	SOS	h-BOASOS
X_1	m	3.3008	3.1059	3.2157	3.278
X_2	m	1.0305	1.1143	0.9986	1.3088
X_3	m	0.6284	0.4289	0.6064	0.6867
X_4	m	0.2	0.2	0.2	0.2
X_5	m	0.5910	0.6348	0.6228	0.6879
X_6	m^2	0.0015	0.0003	0.0271	0.0003
X_7	m^2	0.00070	0.0008	0.0011	0.0007
X_8	m^2	0.001139	0.0008	0.0231	0.0007
X_9	m		3.4673	5.2457	2.9598
X_{10}	m		0.3	0.3	0.3
X_{11}	m		0.3	0.3	0.3
X_{12}	m		0.0004	0.0004	0.0004
Volume of concrete	m^3/m	4.5602	2.5106	3.98298	3.6495
Weight of steel	kg/m	269.296	8.6078	41.2017	73.3119
Min. cost	Rs/m	52589	1.2667e+09	34335.9510	33595.3482

Fig. 7 Schematic diagram of Gear Train Design Problem

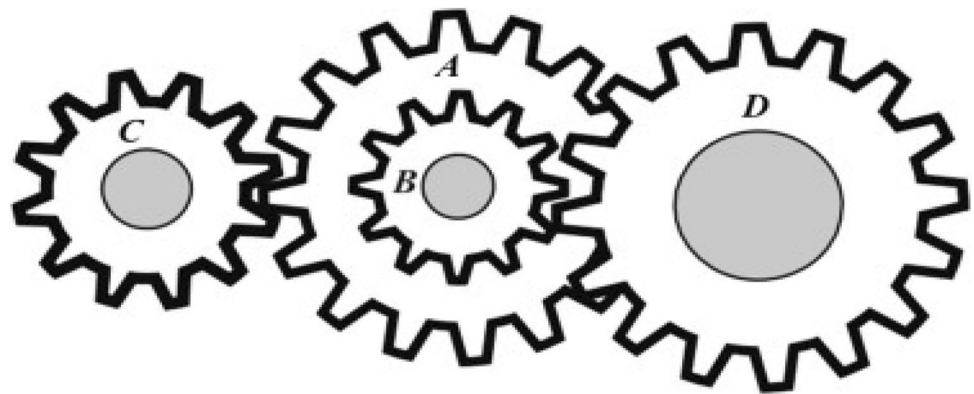
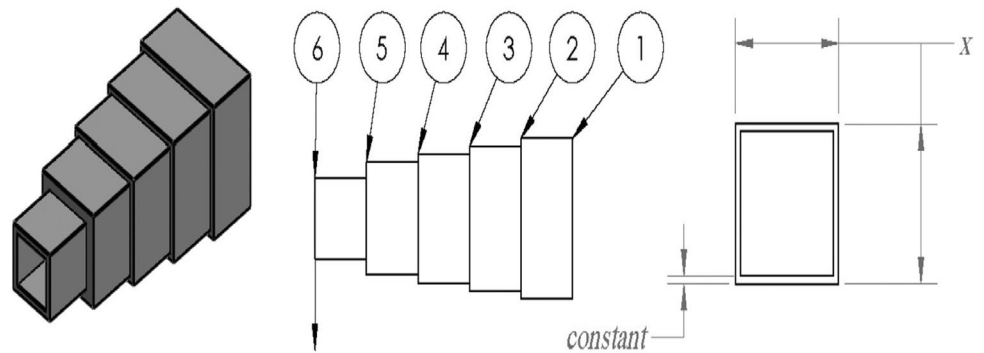


Fig. 8 Cantilever beam design



Acknowledgements The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their useful remarks and important inputs for improving the manuscript.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Dede T (2018) Jaya algorithm to solve single objective size optimization problem for steel grillage structures. *Steel Compos Struct* 26(2):163–170. <https://doi.org/10.12989/scs.2018.26.2.163>
- Mohammad RB, Morteza M (2019) Application of an improved genetic algorithm for optimal design of planar steel frames. *Period Polytech Civ Eng* 63(1):141–151. <https://doi.org/10.3311/PPci.13039>
- Kaveh A, Mohammad F (2019) Optimal seismic design of steel plate shear walls using metaheuristic algorithms. *Period Polytech Civ Eng* 63(1):1–17. <https://doi.org/10.3311/PPci.12119>
- Kaveh A, Seyed RHV, Pedram H (2019) Performance of the modified dolphin monitoring operator for weight optimization of skeletal structures. *Period Polytech Civ Eng* 63(1):30–45. <https://doi.org/10.3311/PPci.12544>
- Ali K, Mohammad ZK, Mahdi B (2019) Optimal design of multi-span pitched roof frames with tapered members. *Period Polytech Civ Eng* 63(1):77–86. <https://doi.org/10.3311/PPci.13107>
- Ahmadi NB, Varaee H Optimal design of reinforced concrete retaining walls using a swarm intelligence technique. *Proc 1st Int Conf Soft Comput Technol Civil, Structural Environ Eng Civ Comp Press, Stirlingshire, UK*
- Khajehzadeh M, Taha MR, El-Shafie A, Eslami M (2010) Economic design of retaining wall using particle swarm optimization with passive congregation. *Aust J Basic Appl Sci* 4(11):5500–5507
- Khajehzadeh M, Taha M R, El-Shafie A, Eslami M Modified particle swarm optimization for optimum design of spread footing and retaining wall. *J Zhejiang Univ-Sci A (Appl Physics Eng)* 011; 12(6):415–4272. <https://doi.org/10.1631/jzus.A1000252>
- Khajehzadeh M, Eslami M (2011) Gravitational search algorithm for optimization of retaining structures. *Indian J Sci Technol* 5(1):1821–1827. <https://doi.org/10.1007/s12205-017-1627-1>
- Yepes V, Alcalá A, Perea C, Gonzalez-Vidoso F (2008) A parametric study of optimum earth-retaining walls by simulated annealing. *Eng Struct* 30(3):821–830. <https://doi.org/10.1016/j.engstruct.2007.05.023>
- Ghazavi M, Bonab SB (2011) Learning from ant society in optimizing concrete retaining walls. *J Technol Educ* 5(3):205–212
- Kaveh A, Abadi ASM (2010) Harmony search based algorithm for the optimum cost design of reinforced concrete cantilever retaining walls. *Int J Civ Eng* 9(1):1–8
- Kaveh A, Behnam AF (2013) Charged system search algorithm for the optimum cost design of reinforced concrete cantilever retaining walls. *Arab J Sci Eng* 38(3):563–570
- Camp CV, Akin C (2012) Design of retaining walls using big bang-big crunch optimization optimum design of cantilever retaining walls. *J Struct Eng* 138:438–448
- Gandomi AH, Kashani AR, Mousavi M (2015) Boundary constraint handling affection on slope stability analysis. *Engineering and Applied Sciences Optimization 2015*. Springer International Publishing, Switzerland 341–358
- Nama S, Saha AK, Ghosh S (2015) Parameters optimization of geotechnical problem using different optimization algorithm. *Geotech Geol Eng* 33(5):1235–1253. <https://doi.org/10.1007/s10706-015-9898-0>
- Nabeel AJ, Ahmed MAY (2016) Optimum Design of Tied Back Retaining Wall. *Open Journal of Civil Engineering* 6:139–155. <https://doi.org/10.4236/ojce.2016.62013>
- Kumar VN, Suribabu CR (2017) Optimal design of cantilever retaining wall using differential evolution algorithm. *Int J Optim Civ Eng* 7(3):433–449
- Bath GS, Dhillon JS, Walia BS (2018) Optimization of geometric design of retaining wall by differential evolution technique. *Int J Comput Eng Res* 8(6):67–77
- Neda M, Sadjad G, Vagelis P (2019) Cost-based optimum design of reinforced concrete retaining walls considering different methods of bearing capacity computation. *Journal* 7:1232. <https://doi.org/10.3390/math7121232>
- Nama S, Saha AK, Ghosh S (2017) Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c- ϕ backfill. *Appl Soft Comput* 52:885–897
- Prayogo D, Cheng MY, Wu YW, Tran D-H (2020) Combining machine learning models via adaptive ensemble weighting for prediction of shear capacity of reinforced-concrete deep beams. *Eng Comput* 36:1135–1153. <https://doi.org/10.1007/s00366-019-00753-w>
- Saha A, Saha AK, Ghosh S (2018) Pseudodynamic bearing capacity analysis of shallow strip footing using the advanced optimization technique “hybrid symbiosis organisms search algorithm” with numerical validation. *Adv Civ Eng*. <https://doi.org/10.1155/2018/3729360>
- Stefanos S, George K, Nikos DL (2020) Conceptual design of structural systems based on topology optimization and prefabricated components. *Comput Struct* 226:106–136. <https://doi.org/10.1016/j.compstruc.2019.106136>
- Lohar G, Sharma S, Saha AK, Ghosh G (2020) Optimization of geotechnical parameters used in slope stability analysis by metaheuristic algorithms. *Appl IoT* 223–231
- Francisco JM, Fernando GV, Antonio H, Victor Y (2010) Heuristic optimization of RC bridge piers with rectangular hollow sections. *Comput Struct* 88(5–6):375–386. <https://doi.org/10.1016/j.compstruc.2009.11.009>
- Degertekin SO, Hayalioğlu MS (2013) Sizing truss structures using teaching–learning-based optimization. *Comput Struct* 119:177–188. <https://doi.org/10.1016/j.compstruc.2012.12.011>
- Nama S, Saha AK, Saha A (2020) The hDEBSA global optimization method: a comparative study on CEC2014 test function and application to geotechnical problem. *Bio-inspir Neurocomput* 225–258
- Gordan B, Koopialipour M, Clementking A, Tootoonchi H, Mohamda ET (2019) Estimating and optimizing safety factors of retaining wall through neural network and bee colony techniques. *Eng Comput* 35:945–954. <https://doi.org/10.1007/s00366-018-0642-2>
- Kaveh A, Biabani Hamedani K, Zaerrezza A (2020) A set theoretical shuffled shepherd optimization algorithm for optimal design of cantilever retaining wall structures. *Eng Comput*. <https://doi.org/10.1007/s00366-020-00999-9>
- Koopialipour M, Murlidhar BR, Hedayat A, Armaghani DJ, Gordan B, Mohamad ET (2020) The use of new intelligent techniques in designing retaining walls. *Eng Comput* 36:283–294. <https://doi.org/10.1007/s00366-018-00700-1>
- Kumar S, Tejani GG, Mirjalili S (2019) Modified symbiotic organisms search for structural optimization. *Eng Comput* 35:1269–1296. <https://doi.org/10.1007/s00366-018-0662-y>
- Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23:715–734

34. Sharma S, Saha AK (2019) m-MBOA: a novel butterfly optimization algorithm enhanced with mutualism scheme. *Soft Comput.* <https://doi.org/10.1007/s00500-019-04234-6>
35. Guo Y, Liu X, Chen L (2020) Improved butterfly optimisation algorithm based on guiding weight and population restart. *J Exp Theor Artif Intell* 1–19
36. Sharma S, Saha AK, Nama S (2020) An Enhanced Butterfly Optimization Algorithm for Function Optimization. In: Pant M, Kumar Sharma T, Arya R, Sahana B, Zolfagharinia H (eds) *Soft Computing: Theories and Applications* (2020). *Advances in Intelligent Systems and Computing*, vol 1154. Springer, Singapore. https://doi.org/10.1007/978-981-15-4032-5_54
37. Sharma S, Saha AK, Ramasamy V, Sarkar JL, Panigrahi CR (2020) hBOSOS: an ensemble of butterfly optimization algorithm and symbiosis organisms search for global optimization. In: Pati B, Panigrahi C, Buyya R, Li KC (eds) *Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing* (2020), vol 1089. Springer, Singapore. https://doi.org/10.1007/978-981-15-1483-8_48
38. Sharma S, Saha AK, Majumder A, Nama S MPBOA - A novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimed Tools Appl*
39. Cheng MY, Prayogo D (2014) Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Computers & Structures* 139:98–112
40. Nama S, Saha AK, Ghosh S (2016) Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decis Sci Lett* 5(3):361–380
41. Nama S, Saha AK, Ghosh S (2017) A hybrid symbiosis organisms search algorithm and its application to real world problems. *Mem Comput* 9(3):261–280
42. Nama S, Saha AK (2018) An ensemble symbiosis organisms search algorithm and its application to real world problems. *Decis Sci Lett* 7(2):103–118
43. Nama S, Saha AK, Sharma S (2020) A novel improved symbiotic organisms search algorithm. *Comput Intell.* <https://doi.org/10.1111/coin.12290>
44. Ezugwu AE, Prayogo D (2019) Symbiotic organisms search algorithm: theory, recent advances and applications. *Expert Syst Appl* 119:184–209
45. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 4:341–359
46. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the 1995 IEEE international conference on neural networks* 1942–8
47. Basturk B, Karaboga D (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. In: *Proceedings of the IEEE swarm intelligence symposium* 12–14
48. Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7(1):19–34
49. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
50. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
51. Yang XS, Deb S (2009) Cuckoo search via lévy flights. *World congress on nature and biologically inspired computing, NaBIC 2009; IEEE*, 210–214
52. Yang XS (2009) Firefly algorithm, levy flights and global optimization. Bramer M, Ellis R, Petridis M (eds) *Research and development in intelligent systems XXVI 2009; Springer, Berlin*, 209–218
53. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3(2):95–99
54. Wang GG, Deb S, Cui Z (2009) Monarch butterfly optimization. *Neural Comput Appl* 24(3–4):853–871. <https://doi.org/10.1007/s00521-015-1923-y>
55. Chakraborty S, Saha AK, Sharma S, Mirjalili S, Chakraborty R (2020) A novel enhanced whale optimization algorithm for global optimization. *Comput Ind Eng.* 107086, ISSN 0360-8352. <https://doi.org/10.1016/j.cie.2020.107086>
56. Wang Z, Luo Q, Zhou Y-Q (2020) Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems. *Eng Comput.* <https://doi.org/10.1007/s00366-020-01025-8>
57. Mirjalili S (2015) The Ant Lion optimizer. *Adv Eng Softw* 83:80–98
58. Sable KS, Patil AA (2012) Optimization of retaining wall by using optintool matlab. *Int J Eng Res Technol* 01(06):1–11

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.