Engineering with Computers
https://doi.org/10.1007/s00366-018-0662-y

ORIGINAL ARTICLE

Modified symbiotic organisms search for structural optimization

Sumit Kumar[1] · Ghanshyam G. Tejani[2] · Seyedali Mirjalili[3]

Received: 1 October 2018 / Accepted: 12 November 2018

Abstract
The structural dynamic response predominantly depends upon natural frequencies which fabricate these as a controlling parameter for dynamic response of the truss. However, truss optimization problems subjected to multiple fundamental frequency constraints with shape and size variables are more arduous due to its characteristics like non-convexity, non-linearity, and implicit with respect to design variables. In addition, mass minimization with frequency constraints are conflicting in nature which intricate optimization problem. Using meta-heuristic for such kind of problem requires harmony between exploration and exploitation to regulate the performance of the algorithm. This paper proposes a modification of a nature inspired Symbiotic Organisms Search (SOS) algorithm called a Modified SOS (MSOS) algorithm to enhance its efficacy of accuracy in search (exploitation) together with exploration by introducing an adaptive benefit factor and modified parasitism vector. These modifications improved search efficiency of the algorithm with a good balance between exploration and exploitation, which has been partially investigated so far. The feasibility and effectiveness of proposed algorithm is studied with six truss design problems. The results of benchmark planar/space trusses are compared with other meta-heuristics. Complementarily the feasibility and effectiveness of the proposed algorithms are investigated by three unimodal functions, thirteen multimodal functions, and six hybrid functions of the CEC2014 test suit. The experimental results show that MSOS is more reliable and efficient as compared to the basis SOS algorithm and other state-of-the-art algorithms. Moreover, the MSOS algorithm provides competitive results compared to the existing meta-heuristics in the literature.

Keywords Natural frequency · Truss optimization · Meta-heuristics · Adaptive mechanism · Exploration · Exploitation

1 Introduction

The design of optimum structure is an active research area due to its wide range of applications in bridges, towers, pylons, roof supports, building exoskeletons, space structures, and industries. Basically, truss is a framework of flexible members pin connected at joint called node intended to withstand purely axial loads over a large span of space. Formerly, structural optimization was a concept of optimal design of load-carrying mechanical structures only. In emerging world, structural optimization is a broad concept due to diversity of aim and possible constraints, causes tremendous variation of results. Basically, objective of structural optimization is to make structures slender with minimum weight. Consequently, optimal design became a progressive area of research in search and optimization field today.

In most of dynamic truss optimization problems, the response highly depends upon the natural frequency and mode shape of structure. Therefore, to avoid condition like resonance, frequency constraints become inevitable in formulation of truss optimization problems subjected to dynamic loads. Mass optimization is considered to be challenging with inclusion of frequency constraints. Frequency

✉ Ghanshyam G. Tejani
p.shyam23@gmail.com

Sumit Kumar
sumit21sep1990@gmail.com

Seyedali Mirjalili
ali.mirjalili@gmail.com

[1] Department of Mechanical Engineering, GPERI, Gujarat Technological University, Ahmedabad, Gujarat, India

[2] Department of Mechanical Engineering, School of Technology, GSFC University, Vadodara, Gujarat, India

[3] School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia

Published online: 27 November 2018

constraints usually conflicts with lower bounded mass optimization, [16]. In addition, regarding design variables, frequency constraints are more implicit, non-linear, and non-convex, [81]. Therefore, gradient-based methods are inappropriate for these types of optimization problems as they have complex programming approach and required additional time for computation. Furthermore, for optimal solution they required good starting point, Kaveh and Talatahari [25]. Due to premature convergence and above demerits, drift is towards nature-based meta-heuristics (MHs) for scholars nowadays as they have leverage with respect to classical MHs. Noilublao and Bureerat [38] proposed an integrated design technique for simultaneously sizing, topology, and shape optimization of truss structure with frequency constraints. Results showed population-based incremental learning MHs are better for natural frequency constrained optimization problems. Gomes [16] attempted to overcome non-linearity associated with shape and size optimization of truss by accounting the particle swarm optimization (PSO) algorithm as an optimization engine with frequency constraints.

Kaveh and Zolghadr [26] suggested an algorithm termed as Charged System Search with its modified version for optimization of structural truss including frequency constraints. Miguel and Miguel [36] suggested the use of two MHs, Firefly Algorithm and Harmony Search, for solving shape and size optimization of trusses simultaneously subjected to multiple natural frequency constraints. Furthermore, [27] introduced a concept of hybridization between the Charged System Search and Big Bang-Big Crunch algorithms with natural frequency constraints, which are proficient to identify local optima trap. In addition, [28] investigated the topology optimization of structural truss with inclusion of buckling, displacement, stress, and frequency constraints. Moreover, [29] comes up with a new algorithm called as Democratic PSO for structural optimization with inclusion of frequency constraints to alleviate the premature convergence phenomenon of the basic PSO algorithm. Farshchin et al. [12] proposed a multi-class Teaching–Learning-Based approach (MC-TLBO) with frequency constraints for structure optimization to increases the exploration capability and enhancing the search efficiency.

Kaveh and Zolghadr [30] developed a multi-agent meta-heuristic (MH) called Tug of War Optimization to optimize size and shape of structural trusses subjected to frequency constraints. Kaveh and Ghazaan [20] employed physically inspired non-gradient algorithm called Vibrating Particles System (VPS) for structural optimization subjected to frequency constraints. Kaveh and Mahdavi [23] utilized colliding bodies' optimization (CBO) for truss structure optimal design subjected to dynamic constraints. Furthermore, Kaveh and Mahdavi [24] proposed two-dimensional version of CBO, with advancement in its algorithm formulation which remain untouched in one-dimension CBO. The results show better search performance within minimum computational time. Savsani et al. [42] comes up with an improved form of the Teaching–Learning-Based Optimization algorithm for optimization of trusses topology considering different static and dynamic constraints. Wang et al. [53] presented an algorithm for 3-D truss structure optimization with multiple constraints on its natural frequencies. Wei et al. [54] proposed a niche hybrid parallel genetic algorithm (NHPGA) for simultaneous size and shape optimization of structural trusses subjected to numerous frequency constraints. Here, NHPGA combined the advantage of simplex search, parallel computing and genetic algorithm with niche technique for getting effective and efficient solution. Zuo et al. [59] for obtaining global solution and to speed up convergence of truss optimization problems used genetic algorithm and hybrid optimality criterion algorithm with multiple frequency constraints.

Kaveh et al. [21] applied Dolphin Echolocation algorithm for optimization of truss structure with frequency constraints. Efficiency of algorithm was compared with hybrid MHs. Kaveh and Ghazaan [83] applied hybridized algorithm for structural optimization with multiple natural frequency constraints. They applied Aging Leader and Challengers with PSO and harmony-based search mechanism with PSO for optimization search problem with natural frequency constraints. Due to its auto tuned capability it has high convergence ability with better stability of structure. Mortazavi and Togan [37] demonstrate the competitive edges of integrated PSO in optimization of truss structures with integration of the weighted particle concept and fly-back mechanism under multiple frequency constraints with sizing and layout variables. Savsani et al. [43] considered four modified MHs to solve distinct obstacles of optimization.

Recently many MHs showed the superiority with respect to gradient-based algorithms as they do not require gradient-based calculation [33]. MHs include the natural evolutionary process like the evolutionary algorithm proposed by Fogel [13], De Jong [5], Koza [32] and the Genetic Algorithm (GA) proposed by Goldberg and Holland [15], and animal behavior, e.g., Tabu search proposed by Glover [14], Ant algorithm proposed by Dorigo et al. [7], PSO proposed by Eberhart and Kennedy [8], the Artificial Bee Colony algorithm proposed by Karaboga and Basturk [17], Firefly Algorithm by Yang [56], Cuckoo Search by Yang and Deb [57], Dolphin Echolocation by Kaveh and Farhoudi [18] and the physical annealing process like simulated annealing proposed by Kirkpatrick et al. [31] and more like Harmony Search by Geem et al. [78], Big Bang–Big Crunch by Erol and Eksin [9], Water Cycle Algorithm by Eskandar et al. [10], and Ray Optimization by Kaveh and Khayatazad [22]. However, above MHs need extra controlling parameters and also not easy to use. For example, GA needs mutation and crossover rates, while PSO requires inertia weight, social, and cognitive parameters. Furthermore, many MHs need to perform parameter tuning, where improper tuning related to algorithm-specific parameters might increase computational time and produce local optima solutions. They show inadequacy in robustness and generalization.

The resemblance in MHs is that they incorporate rules and randomness to imitate organic phenomena. MHs generally employs various iterative process to find efficient optimal solution in search space with integration of exploration and exploitation notion to escort a minor heuristic with learning strategies [39]. However, Wolpert and Macready [55] rationalized that it is impossible to solve all optimization problem with one MH. Therefore, a recent efficient and high performing MH is required to solve complex optimization problems.

Symbiotic Organisms Search (SOS) is a recent algorithm which mimics the strategies of mutualism, commensalism, and parasitism encouraged by natural synergy between organisms in ecosystems. SOS has better exploitation capability across benchmark problems due to ease in adjustability of its common parameters which make it simple to operate as well as its ability of merit solution generation within fewer iterations [3]. Different studies acknowledged the novelty of SOS than other competing MHs due to its simplicity of implementation with ability of better exploration and exploitation using minimum controlling parameters. Many scholars also compared its performance for various benchmark functions and optimization problems across other MHs with outcome of better performances [60–64]. Moreover, lots of research demonstrated the superiority of SOS over other MHs like Panda and Pani [65] united the SOS with Augmented Lagrange Multiplier method to solve constrained optimization problems. This fusion enhances the result accuracy within lower run time compared to other MHs. Prayogo et al. [66] applied SOS with an improvement in the parasitism phase for optimization of resource leveling of construction project. The experimental result shows the better quality solution in comparison with existing optimization models. Similar ascendancy conduct of SOS has been detected by Yu et al. [58] while application in capacitated vehicle routing problem. Authors proposed six improved version of SOS with inclusion of two new interaction strategies, namely competition and amensalism in the basic SOS algorithm for performance enhancement. Likewise, a new hybrid artificial intelligence system, SOS–LSSVR (least squares support vector regression), has been proposed to predict the permanent deformation potential of asphalt pavement mixtures (Cheng et al. [4]). Proposed study is able to achieve better accuracy than all other comparative measures which manifest the outperformance of SOS-based system over other methods. Abdullahi et al. [63] presents a Discrete Symbiotic Organism Search (DSOS) for optimal scheduling of tasks on cloud resources. Simulation results revealed that DSOS outperforms PSO which is one of the most popular heuristic optimization techniques used for task scheduling problems. DSOS converges faster when the search gets larger which makes it suitable for large-scale scheduling problems. Banerjee and Chattopadhyay [67] proposed a modified SOS to design an improved 3 dimensional turbo code. SOS a novel powerful MH optimization technique is also proposed for the first time to solve the load frequency control problem [68]. Results show that the dynamic stability of the concerned power system effective enhancement with SOS. Furthermore, Prayogo et al. [72] in their doctoral dissertation compared the effectiveness of SOS in solving various civil engineering and benchmark problems. Simulation results demonstrate that SOS is significantly more effective and efficient than the other algorithms present in the literature. Analogously, lots of research [69–71, 73, 74] are available in the literature which flourishes the supremacy of SOS over other MHs.

Few studies demonstrated the supremacy of adaptive strategies over fixed search mechanism for quality solution and convergence search which tune the balance between exploration and exploitation [11, 50]. For improvisation in exploration and exploitation potential of the basic SOS algorithm, Tejani et al. [46] introduced an adaptive benefit factors which result into more-reliable and balanced solution of dynamics structures subjected to frequency limits. Tejani et al. [48] suggested a modified and improved version of SOS to overcome non-convexity, non-linearity drawback of basic algorithms with renovation in search performance. Ezugwu et al. [11] presents SOS with simulated annealing (SA) to solve the traveling salesman problems. Results demonstrate better solution convergence, minimum execution time with minimum deviation of solution from best solution. Yu et al. [58] presented improved SOS with two new interaction strategies called competition and amensalism. Liao and Kuo [35] develop five new discrete SOS algorithms. Tran et al. [51] introduced opposition multiple objective SOS for scheduling repetitive projects. Similarly, Tejani et al. [52] utilized multiobjective adaptive symbiotic organisms search for truss optimization problems. Despite of various advantages, above-mentioned algorithms lacks in key components. In addition, in these algorithms if their exploration capacity is superior then exploitation capacity decreases comparatively and vice versa. Furthermore, they untouched the parasitism phase which leads to problem of large computational time for getting solution of optimization process. Therefore, for better result a good balance between exploration and exploitation capacity of algorithm is required.

Non-linearity and non-convexity characteristics of shape and size variables with frequency limits, cause optimization problem more challenging. SOS incompetency for dynamic structural optimization with natural frequency limits has been displayed by many scholarly findings [79], (Crepinsek et al. 2013). Research depicts the heuristic nature of benefit factor, i.e., 1 or 2 in the basic SOS algorithm, which represents the organism interaction as partial or complete only. However, the benefit factor which organism may get could be in any scale between 1 and 2. The SOS algorithm though efficient in solving complex optimization and discrete engineering problems, still has high probability of plunging into local optimum [48]. The SOS algorithm also lags in iterative process of size of population selection. The reason is there are instances when selecting a small population size yields a better result than a large population size and vice versa. For example, if the algorithm is stopped too early, the approximation of the solution might not be close to the targeted global optimum and prolonging the simulation might as well incur unnecessary scale up in the computational effort. Ezugwu et al. [11] have shown the improvement in efficiency of algorithm via reducing the solution completion time, introduce diversity in the search process and avoid premature convergence through integration of local search strategy into basic SOS. Above researches motivate the researchers to extend their research by incorporating additional improvements into SOS using hybridization techniques, to include other MHs such as SA and test its performance on different set of larger problem instances. In addition, many researches show the requirement of improvement in basic SOS for discrete and continuous optimization problems [6]. Few research scholars show the possibility of better trade of between exploration and exploitation capability of SOS with modification in mutualism and parasitism phase. Miao et al. [84] presents a modified SOS algorithm based on the simplex method to solve the path planning problems. Results shows, modification of SOS leads to faster convergence speed, higher precision, and stronger robustness than the main algorithm. Numerous scholars show that metamorphosis lead to increase in the diversity of the population and improvement in the ability of the algorithm to explore and exploit, as well as preventing the algorithm from prematurely finding the local optimal solution. Moreover, modification testing on various benchmark problems shows enhancement in accuracy and robustness with stronger convergence speed than other algorithms.

Despite of aforementioned advantages, the SOS algorithm inadequacy for dynamic structural optimization with natural frequency limits has been uncovered by many scholarly findings. Moreover, it is practically not possible to forecast the influence of the adaptive operators for different real life applications [49]. As per the No Free Lunch theorem, it is impossible to solve all optimization problems with one algorithm. Furthermore, the SOS algorithm is a recently developed algorithm and it is always interesting to investigate different modification that can improve the performance of the algorithm. These proficiencies and prospects encouraged us to formulate an adaptive SOS algorithm with frequency constraints and to investigate its effect on dynamic structural optimization problems. These inadequacies like lack of efficient exploration, balanced search with good success rate and less computation time, give impetus to formulate a modified SOS (MSOS) algorithm and to examine its effect on different structural optimization problems.

In other words, a good balance of exploration and exploitation is essential to avoid local solutions and find an accurate estimation of the global optimum for a given optimization problem. To alleviate these drawbacks, MSOS is equipped with adaptive benefit factor to tune the balance between exploration and exploitation during mutualism phase and an improved parasitism phase to boost exploitation capability during parasitism phase of SOS.

The rest of the paper is organized as follows: Sect. 2 presents the SOS algorithm. The improved SOS is proposed in Sect. 3. The problem is formulated in Sect. 4 and solved using the modified SOS in Sect. 5. Moreover, Sect. 6 includes verification of MSOS performance with various functions of CEC 2014 test suits. Finally, Sect. 7 concludes the work and suggest future directions.

2 The symbiotic organisms search algorithm

In nature, an essential relationship is maintained by many organisms for survival and growth. These relations may be beneficial or harmful sometimes. In symbiotic relationship species live together for mutual benefit and survival, which make it different from other regular interaction. Considering the natural interaction between the organism for survival in ecosystem Cheng and Prayogo [3] introduced a compelling and modest meta-heuristic algorithm called SOS which simulates the interactive behavior among organism seen in nature. Symbiosis describes close interaction between two or more species like snail and tortoise, human and dog, Lichen (algae and fungus), etc. In either of case, each organism helps together in existence and continuation while prevailing relationship.

Unlike other MHs which need extra optimization parameters SOS algorithm required only general controlling parameters like total no of function evaluations (FE) for its operation and population size (number of organism). For example, GA needs mutation, crossover, and rates, while

PSO requires inertia weight, social, and cognitive parameters [46]. On the other hand, SOS does not need algorithm-specific governing factors and is simply organized and easy to use. It is also robust and generalized. This is considered an advantage over competing algorithms, because SOS does not need to perform parameter tuning. Improper tuning related to algorithm-specific parameters might increase computational time and produce local optima solutions.

Symbiotic relationship can be divided into three phases: mutualism, commensalism and parasitism.

Mutualism: In this phase, the organism tries to find the best solution in mutual benefit interaction with other to find out the optimum value greater than previous one– a classic example is the interaction between bees and flowers;

Commensalism: In this phase, organisms with best functional value is trying to find without affecting the other possible solution (organism) nearby– an example is the relationship between remora fish and sharks; and

Parasitism: In this phase, best solution is obtained with suffering of other organism– an example is the plasmodium parasite, which uses its relationship with the anopheles mosquito to transfer between human hosts.

The three phases are adopted from the most common symbioses used by organisms to increase their fitness and survival advantage over the long term. In this optimization algorithm, the better solution can be achieved by the symbiotic relations between the current solution vector and either the other randomly selected solution or the best solution from the current population. The phases are repeated until the stopping criterion is achieved.

The detailed discussion of all three phases and modification of SOS is explained in the subsequent sections:

2.1 The mutualism phase

When both organisms of different species get individual benefit from interaction then it is called a mutualism. A common example of mutualism is the relationship between bees and flowers, where both organisms acquire benefits from each other. Bees obtained sweet food source secreted by the flowers (called nectar) through visiting different flowers which benefit them to produce honey. On the other hand, while travelling bees transfer pollen grains from one plant to another which facilitate pollination in flowers (benefit to flowers).

In mutualism phase, a sequentially selected organism (X_i) in the ecosystem interacts with another random member say (X_j), where $i \neq j$. Both organisms engage in a mutuality relationship with the goal of increasing mutual survival advantage in the ecosystem. They are interdependent and each gain benefit from other. New candidates are calculated based on interact with the motive of mutual survival advantage in the ecosystem as per Eqs. (1, 2):

$$X_{\text{inew}} = X_i + \text{rand}(0,1) \times (X_{\text{best}} - \text{Mutual\_Vector} \times BF_1), \tag{1}$$

$$X_{\text{jnew}} = X_j + \text{rand}(0,1) \times (X_{\text{best}} - \text{Mutual\_Vector} \times \text{BF}_2) \tag{2}$$

where

$$\text{Mutual\_Vector} = \frac{X_i + X_j}{2} \tag{3}$$

$$\text{BF}_1 = 1 + \text{round}[\text{rand}(0,1)]. \tag{4}$$

$$\text{BF}_2 = 1 + \text{round}[\text{rand}(0,1)] \tag{5}$$

The following observations on the mutualism mathematical model can be made:

- rand(0,1) in Eqs. (1, 2) is a vector resenting random numbers between 0 and 1;
- Mutual_Vector in Eq. (3) represents the relationship attributes between organisms X_i and X_j;
- X_{best} represents the organism with the current highest state of adaptation to the ecosystem;
- Organism X_i might benefit significantly when interacting with organism X_j; at the same time, organism X_j might benefit only slightly when interacting with organism X_i.

2.2 The commensalism phase

Living organisms in nature generally react and live with one another for adaptation in various forms. Commensalism is an interesting relationship one out of them. In this type of symbiotic relationship, one will get benefit, while the other will remain unaffected or not harmed. Some time it is also called as one-sided symbiotic relationship. The welfare can be in any form like seed dispersal, transportation, food, and shelter. One of the examples is remora and shark in which remora remains attached with shark for getting leftover food but shark remain neutral.

In this phase, the best from mutualism phase taken as an input and an organism is arbitrarily selected from ecosystem which tries to communicate with others. Unlike mutualism phase, modification is only for one organism and other remain as previous one, whereas overhaul is only when new fitness value is superior to previous. The new organism is presented in Eq. (6):

$$X_{\text{inew}} = X_i + \text{rand}(-1,1) \times (X_{\text{best}} - X_j) \tag{6}$$

Some observations on the commensalism mathematical model can be made:

• rand(−1, 1) in Eq. (6) is a vector of random numbers between − 1 and 1;

• X_{best} reflects the current highest state of adaptation to the ecosystem, similar to that used in the mutualism phase;

• Organism X_i is updated to X_{inew} only if its new fitness is better than its pre-interaction fitness; and

• For each organism X_i, this interaction counts for one function evaluation.

2.3 The parasitism phase

In parasitism phase, parasite gets only benefit while living in host body, causing harm to host from its long-term interaction. Here one organism gets complete benefit, while other gets eliminated completely from the system. One of the examples is plasmodium parasite which use Anopheles mosquito to enter into human body which act like host. The parasite flourishes and reproduces inside of host and human body suffers disease which may result into death.

After the commensalism phase is completed, the organism X_i again randomly selects a new organism from the ecosystem, organismX_j. In this phase a design vector (X_i) of "i" population organism play role of anopheles mosquito. Here anopheles mosquito creates an artificial vector called "Parasite Vector". During the interaction, Parasite Vector tries to kill host X_j and replace it in the ecosystem. Organism X_i may gain a benefit, because, by cloning Parasite Vector, its influence in the ecosystem may increase, whereas X_j may suffer and die.

The Parasite Vector is presented in Eq. (7):

$$\text{Parasite Vector} = \begin{cases} X_j & \text{if } 0.5 \le \text{rand}(0,1) \\ \text{LB} + \text{rand}(0,1) \times (\text{UB} - \text{LB}) & \text{Otherwise} \end{cases} \tag{7}$$

where LB and UB represents the solution lower bound and upper bound, respectively.

The creation of Parasite Vector is described as follows:

1. An initial Parasite Vector is created in the search space by duplicating organism X_i; some decision variables from the initial Parasite Vector are modified randomly to differentiate Parasite Vector from organism X_i;
2. A random number is created within a range from one to the number of decision variables, representing the total number of modified variables;
3. The location of the modified variables is determined stochastically using a uniform random number, which is generated for each dimension; if the random number is less than 0.5, the variable is modified; otherwise, it stays the same; and
4. The variables are modified using a uniform distribution within the search space and Parasite Vector is ready for the parasitism phase.

Both Parasite Vector and organism X_j are then evaluated to measure their fitness. If Parasite Vector has a better fitness value, it kills organism Xj and assumes its position in the ecosystem. If thefitness value of X_j is better, X_j has immunity from the parasite and Parasite Vector can no longer live in that ecosystem. For each organism X_i, this interaction counts for one function evaluation.

2.4 Stopping criterion

The optimization process terminates when a user-set stopping criterion is met. This criterion is often set as the maximum iteration number ($g_{\max}$) or the maximum number of function evaluations ($FE_{\max}$). The optimal solution can be identified after search process termination.

3 Modification in the SOS algorithm

Success rate of most of the population-based optimization algorithms depends on the equilibrium between exploration and exploitation abilities significantly. Exploration represents the ability of algorithm for global search that strongly affects the accuracy of obtained optimal solution. Whereas exploitation signifies the local search potential, this plays an important role in impacting on the convergence of the optimization algorithm. Obviously, if the exploration capacity is superior to the other, a global optimal solution can be achieved, yet the convergence is slow. This is due to the fact that the algorithm must require a remarkable amount of computational cost for seeking an optimal solution in a given whole domain. Conversely, the algorithm converges quickly, but optimal solutions may occur. Therefore, provided that the above two abilities are adjusted to gain a better balance, the solution accuracy and the convergence speed can be obtained simultaneously. Although the original SOS algorithm is good at the global search capability, the limitation on the computational cost is survived. As observed, both mutualism and commensalism phases improve the exploitation ability of the algorithm, while the parasitism phase contributes to the exploration capability. Nevertheless, the last phase requires a considerable amount of computational cost to sever its search process. If the parasitism phase is eliminated to save computational cost, the SOS approach is easily trapped at local solutions. Therefore, the other phases must be refined to preserve the balance between exploitation and exploration capabilities.

To overcome the aforementioned imperfections of the original SOS approach, a MSOS algorithm is proposed in this study.

3.1 Modification in mutualism phase

Various research applications proved that the structural optimization with adaptive control algorithm has dominance in solving single and multiobjective problems with inclusion of desired properties like good convergence, superior search ability and harmony between exploration and exploitation. Adaptive control mechanisms usually advance the effectiveness of the algorithm and tune the balance between exploration and exploitation abilities [49]. In this respect, many studies have been proposed by employing various adaptive control approaches in various MHs. Piotrowski [41] used the concept of global and local mutation operators to propose an adaptive memetic differential evolution (DE) algorithm, whereas a strategically adaptive version of a DE was proposed by Bureerat and Pholdee [85]. Li and Yin [34] proposed a modified cuckoo search algorithm using an adaptive parameter setting to enhance the diversity of the population. Shan et al. [44] used adaptive control on an artificial bee colony algorithm to improve the performance. Yi et al. (2016) employed an adaptive DE based on fitness function value. Tejani et al. [47] modified a teaching-learning-based optimization using an adaptive teaching factor. Many studies to develop algorithms with adaptation for multiobjective optimization are also found in the literature. Similarly, Tejani et al. [52] utilized multiobjective adaptive symbiotic organisms search for truss optimization problems. Bingul [2] proposed an adaptive GA with dynamic fitness function for multiobjective design problems. Ou-Yang et al. [86] used a self-adaptive-velocity PSO. Pham [87] presented adaptive directional mutation to enhance DE.

On the other hand, SOS is a newly discovered algorithm which encourage investigator to enhance the algorithm performance with further modification in it. Moreover, research shows that it is impossible to find an algorithm which is able to solve all optimization problems individually, so always modification is needed in optimization algorithm [49]. In addition, SOS is new player in this arena, so it always has a good possibility of improvement in its performance.

Research depicts the heuristic nature of benefit factor, i.e., 1 or 2 in Mutualism phase of basic SOS algorithm, which represents the organism interaction as partial or complete only. However, benefit factor is the key component in influencing the Mutual_Vector in mutualism phase of SOS. The benefit factors ($\mathrm{BF}_1 \mathrm{and} \mathrm{BF}_2$) are determined stochastically as either 1 or 2 (Eqs. 4 and 5) in basic SOS algorithm, indicating whether an organism partially or fully benefits from the interaction. These Benefit Factors is heuristic in nature, because in nature one organism can receive partial or full benefit than other. The level of benefit interaction, i.e., full or partial is represented by these factors. Organisms evolve to a fitter version only if their new fitness is better than their pre-interaction fitness; if so, the old X_i and X_j are replaced by X_{inew} and X_{jnew}; this mechanism is similar to greedy selection; and for each organism X_i, this interaction counts for two function evaluations.

In other words, the organism X_i and X_j can get partially and fully benefit from mutual vector. Which means when lower value of benefit factor is there then algorithm search will be fine with small step, but the convergence speed of the algorithm decreases. Similarly, if larger value of benefit factor considered the search get speed up skipping nearby value which reduces exploitation capacity of algorithm. However, the benefit factor which organism actually may get could be in any scale between 1 and 2 in nature. This motivates us for changing benefit factor (BF_1) to adaptive benefit factor (ABF) which gives good convergence, superior search ability, and harmony between exploration and exploitation:

$$X_{\mathrm{inew}} = X_i + \mathrm{rand}(0,1) \times (X_{\mathrm{best}} - \mathrm{Mutual\_Vector} \times \mathrm{ABF}). \tag{8}$$

$$X_{\mathrm{jnew}} = X_j + \mathrm{rand}(0{,}1) \times (X_{\mathrm{best}} - \mathrm{Mutual\_Vector} \times \mathrm{BF}_2) \tag{9}$$

where Mutual_Vector is defined as Eq. 3.

$$\mathrm{ABF} = \begin{cases} f_i(X_i)/f_i(X_{\mathrm{best}}), & \mathrm{if} f_i(X_{\mathrm{best}}) \neq 0 \\ 1 + \mathrm{round}[\mathrm{rand}(0{,}1)], & \mathrm{if} f_i(X_{\mathrm{best}}) \neq 0 \end{cases} \tag{10}$$

$$\mathrm{ABF} = \begin{cases} 1, & \mathrm{if\,ABF} < 1 \\ 2, & \mathrm{if\,ABF} > 2 \\ \mathrm{ABF}, & \mathrm{otherwise.} \end{cases} \tag{11}$$

The adaptive benefit factor as shown in Eqs. (10, 11) is applied for minimization problem. Here value of the design variables in this algorithm may change to a small extent or to a significant extent as they are governed by various factors. The large and small changes in the design variables represent the exploration and exploitation of a search space, respectively. For an accurate optimization of structure with minimization of objective function, a good balance between the exploration and exploitation is needed. Exploration refers to the process of finding promising areas of a search space and leads to global search. Exploitation, however, is the local search around desired solutions found in the exploration phase.

Due implementation of adaptive benefit factor, the term $\mathrm{Mutual_Vector} \times \mathrm{ABF}$gives a good balance between exploration and exploitation for the relationship characteristic between organism X_i and X_j compared with those of initial BF_1 and BF_2 rand(0,1) values. This leads to good diversity with faster convergence and more stability of$(X_{\mathrm{best}} - \mathrm{Mutual_Vector} \times \mathrm{ABF})$. As a result, new solutions of X_{inew} and X_{jnew} lies inside feasible region that minimize the deviation of the objective function between the best organism and the whole ecosystem. Consequently, the stability of the algorithm based on the evaluation of standard deviation increases, and the algorithm demands less numbers of analyses to converge a global optimal solution. Eventually exploration ability of mutualism phase is increase due to implemented modification in Benefit factor.

ABF leads the algorithm to explore non-visited region in search space when population('*i*' or '*j*') is away from the best population and also helps in increase the convergence rate of solution when population is nearby to best solution. This shows that MSOS with ABF leads to the global optimal solution with a good balance between exploration and exploitation of optimization algorithm.

3.2 Modification in parasitism phase

A parasitism phase is important in upgrading the exploration capacity of SOS. However, it is also experienced that over exploration results in higher computational cost. In this phase, a large number of new solutions get rejection due to inferior objective functional values compared to previous one. In parasitism phase of basic SOS algorithm, the exploration rate is poor, as the parasite vector is producing with fusion of design variable with a random generated variable in search space. This only results in improvement of existing result, which enhances the exploitation capability of this algorithm. The main reason for modification is to remove the drawback of low exploitation capability of parasitism phase. Many studies show that the exploitation capability of parasitism phase in SOS algorithm is considerably low as compared to exploratory capability. Increasing the number of *FE* leads to an increase in the convergence time too.

In addition, many researches show the improvement in efficiency of algorithm with modification in parasitism phase [45, 66]. Therefore, this phase is improved with the modification of parasitism phase. Here it is tried to improve exploitation potential of parasitism phase with maintaining global optimal solution as well in search space. Therefore, our motive is to set a perfect balance between exploration and exploitation of search algorithm. In this the proposed algorithm, exploration is encourage using mutation strategy. The strategy enhances the diversity of population and solutions as well. The proposed approach allows the algorithm to explore different regions of the search space at the same time, avoid the population concentration in one region, and avoid premature convergence. Equation 12 represents the modification in parasite vector:

$$\text{Modified Parasitism Vector} = \begin{cases} X_i^j & \text{if } 0.5 \le \text{rand}(0,1) \\ X_k^j & \text{otherwise} \end{cases} \tag{12}$$

Here, the randomly generated organism X_k is replaced by parasitism vector when parasitism vector function value is better than host X_k. This is governed by the condition when a random number in [0, 1] has a greater value than a threshold ($0.5 \le \text{rand}(0, 1)$), where threshold value 0.5 is adapted iteratively. Otherwise new organism '*k*' is selected randomly from the population. Here if the improved parasitism vector is fitter than the organism '*k*', parasite will kill organism '*k*' and acquire its place. Finally, if the objective function value for the improved parasitism vector gives minimum value than the previous one, then the parasitism vector takes the new position while eliminating previous organism.

Populations are evolving to a fitter version only if their new fitness dominates their pre-interaction fitness. In this case, the old X_i will be replaced immediately byX_{inew}. The X_i will be moved into advanced population. Otherwise, X_{inew} will be added into advanced population for selecting the next generation ecosystem. As such, the proposed algorithm can converge faster while maintaining good diversity. Due to above-mentioned modification in parasitism phase exploitation capability increase with high convergence rate and stability of optimal solution.

It should be noted that in this modification commensalism phase kept remain same as in the original SOS algorithm. In commensalism phase, each organism $X_{i\text{new}}$ and $X_{j\text{new}}$ is compared with the pre-interaction organism X_i and X_j to choose a better organism for the next step. Parameter ($X_{\text{best}} - X_j$), of Eq. (6) echoes the leverage getting by X_i due its relationship withX_j. In this phase, an arbitrary organism X_i interacts with randomly selected another organism X_j in ecosystem in which only organism X_i benefited, while other, i.e., X_j remain unaffected. The organism updated only when the function value F (X_{inew}) fitter than previous one. This helps X_i to ameliorate in ecosystem with respect to existing best organism X_{best}. It is clear that the worse organism in each pair is not selected and convergence speed increased with better organism selection. In this phase exploitation ability and convergence speed are improved due to the decreased search space. Therefore, in this phase a good exploitation near the best organism region is observed in search space which accelerates the convergence of algorithm.

However, with the afore-implemented modifications in the mutualism and parasitism phases of MSOS, the exploration and exploitation abilities are balanced. Moreover, the suggested parameter values in the MSOS have faster convergence speed against the original SOS. It can be concluded that MSOS significantly improves the computational cost and the convergence of the original SOS, but still attaining the global optimal solution with high accuracy and reliability. Both implementations aim at increasing the exploration and exploitation capability and thus enhancing the convergence speed with higher precision,

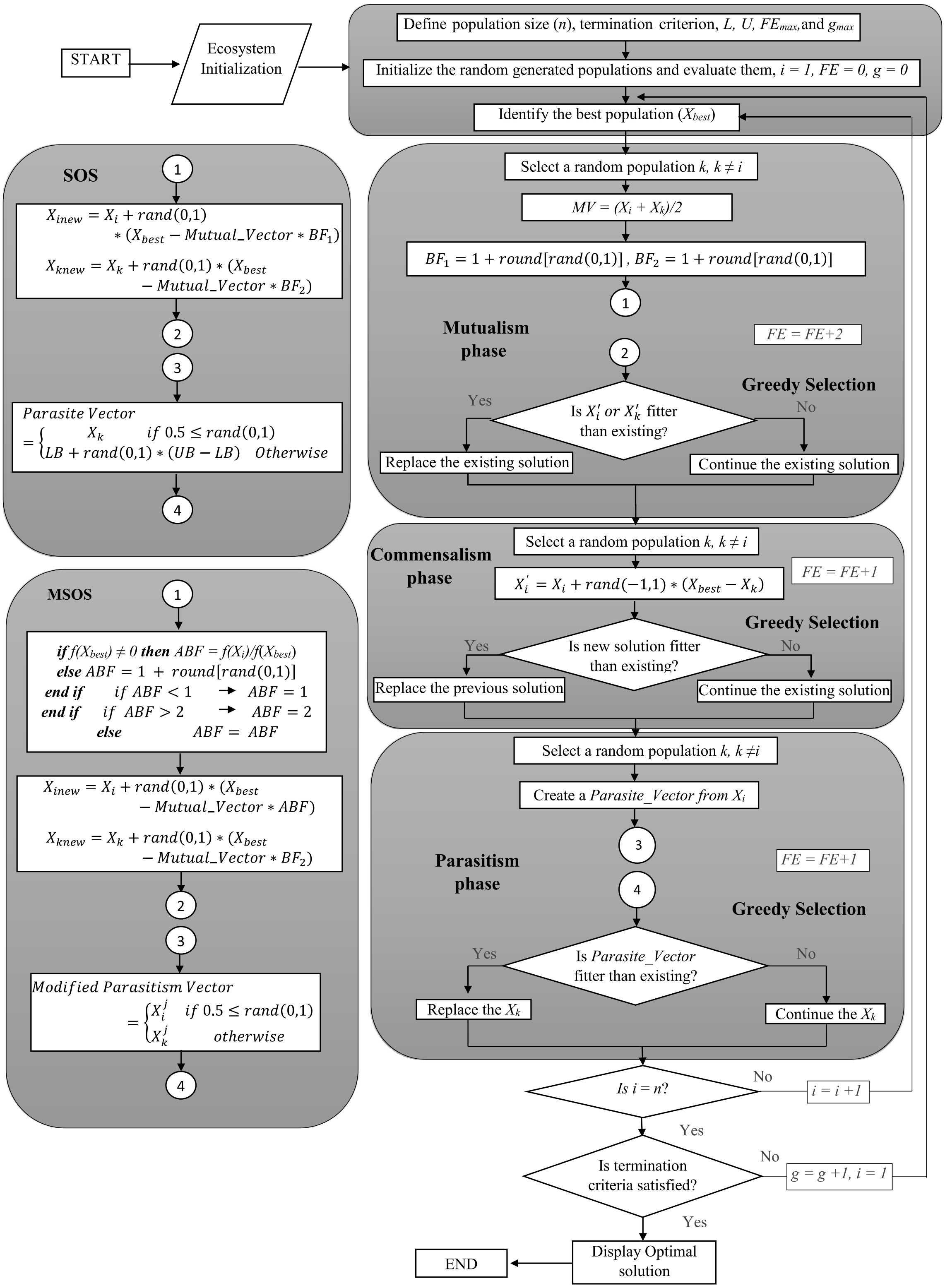

Fig. 1 Flowchart of the SOS and MSOS algorithms

and stronger robustness than basic SOS. From above modification, a balanced trade-off between the exploration and exploitation abilities is, therefore, achieved. Modification increases the diversity of the population and improves the ability of the algorithm to explore and exploit, as well as preventing the algorithm from premature convergence. Furthermore, the proposed algorithm is simple and easy to implement.

The proposed algorithms work in three phases: the mutualism phase, the commensalism phase, and the parasitism phase. In addition, each phase is governed by a number of generations and various factors. A schematic diagram of MSOS and SOS is presented in Fig. 1, which gives various steps of these algorithms like initialization, mutualism phase, commensalism phase, parasitism phase, and termination criteria. The subsequent sections examine the efficiency of MSOS with respect to the shape and sizing problems.

4 The formulation of the design problem

The basic objective of structural optimization problem is to minimize the mass of the truss via finding optimal element cross section area and optimal nodal positions subjected to multiple frequency constraints. Therefore, the objective function is the mass of truss by neglecting lumped mass at nodes while keeping nodal coordinates and elemental cross-section areas as design variables.

The mathematical formulation corresponding to the problem considered in this work is as follows:

$$\text{Find, } X = \{A, N\}, \text{where } A = \left\{A_1, A_2, \dots, A_m\right\} \text{ and } N = \left\{N_1, N_2, \dots, N_n\right\} \tag{13}$$

to minimize, Mass of truss,

$$F(X) = \sum_{i=1}^{m} A_i \rho_i L_i$$

Subjected to

$$g_1(X) : f_q - f_q^{\min} \geq 0$$

$$g_2(X) : f_r - f_r^{\max} \leq 0$$

$$g_3(X) : A_i^{\min} \leq A_i \leq A_i^{\max}$$

$$g_4(X) : N_j^{\min} \leq N_j \leq N_j^{\max}$$

where $i = 1,2, \dots, m; j = 1,2, \dots, n$

where $A_{i,} \rho_{i,}$ and L_i signify the cross-sectional area, material density, and length of the bar 'i', respectively. N_j presents nodal coordinate (x_j, y_j, z_j) of 'jth' node. f_q and f_r are 'qth' and 'rth' natural frequencies, respectively. The superscripts, 'max' and 'min' signify the maximum and minimum allowable limits, respectively. The finite element method is applied as analyzer to calculate fundamental Eigen values and natural frequencies of the truss structure.

Solving constrained optimization problem at times become infeasible as the solutions at times get stuck in the local optima, especially in the problems having disjoint search space. Under these scenarios Penalty function methods are used which convert a constrained problem into an unconstrained one, where the 'Penalty Function' penalizes the infeasible solutions to move toward desirable feasible solutions.

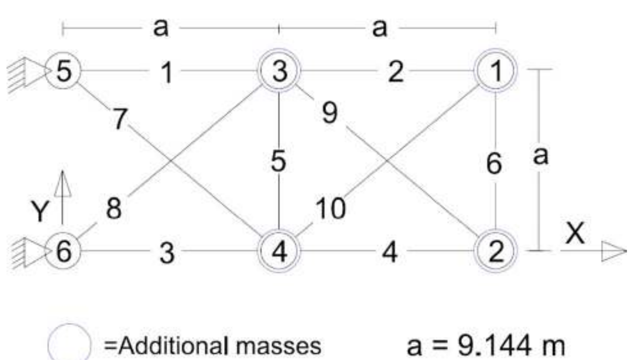

Fig. 2 10-bar truss

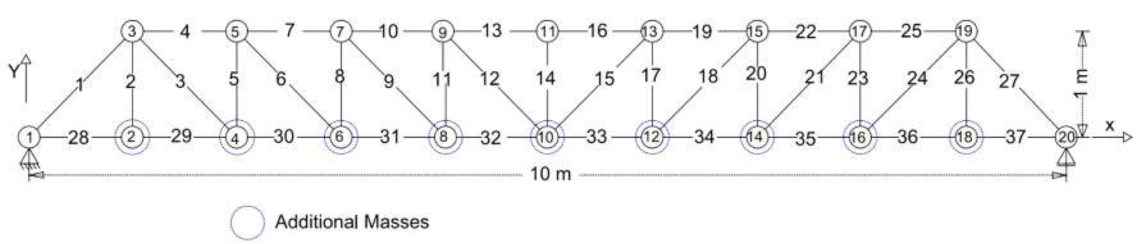

Fig. 3 37-bar truss

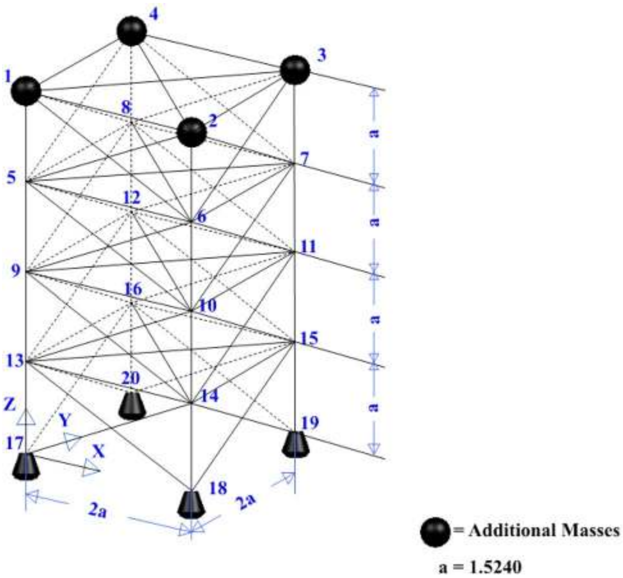

Fig. 4 72-bar truss

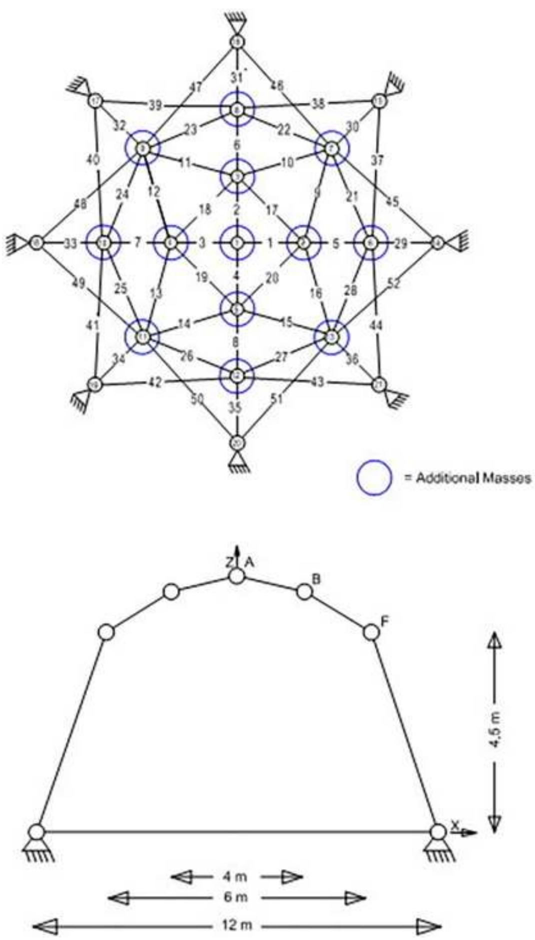

Fig. 5 52-bar truss

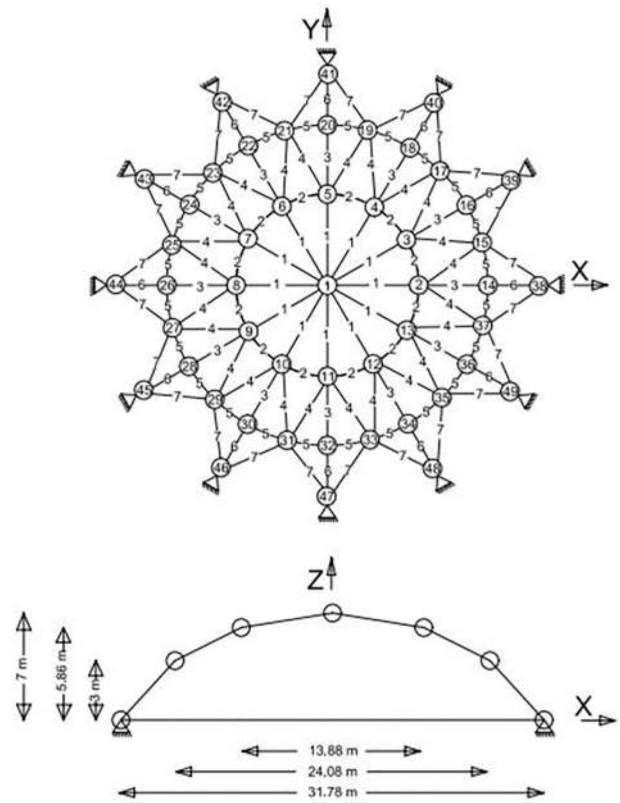

Fig. 6 120-bar truss

The objective function is penalized to handle frequency limits. There is no penalty for non-violation of the limits; otherwise, the penalty function is considered as follows [28]:

$$\text{Penalized } F(X) = \begin{cases} F(X), \text{if no violation of limits} \\ F(X) \times F_{\text{penalty}}, \text{otherwise} \end{cases} \tag{14}$$

$$F_{\text{penalty}} = (1 + \varepsilon_1 \times \mathtt{C})^{\varepsilon_2}, \mathtt{C} = \sum \left(\mathtt{C}_q + \mathtt{C}_r\right),$$
$$\mathtt{C}_q = \left| 1 - \frac{\left| f_q - f_q^{\min} \right|}{f_q^{\min}} \right|, \mathtt{C}_r = \left| 1 - \frac{\left| f_r - f_r^{\max} \right|}{f_r^{\max}} \right| \tag{15}$$

The parameters ε_1 and ε_2 are selected by considering their nature. In this investigation, values of ε_1 and ε_2 are set as 3 as per previous studies [48, 75].

5 Truss problems and discussions

In this section, the proposed algorithm is tested on six challenging benchmark trusses (as shown in Figs. 2, 3, 4, 5, 6, and 7). Here the consideration is to perform size and shape optimization of given trusses under multiple natural frequency limits. Later on, soundness and viability of the proposed

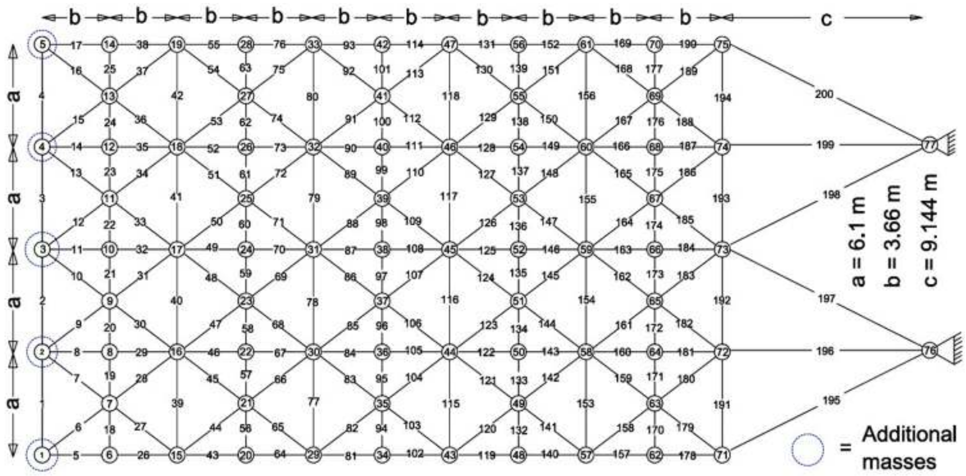

Fig. 7 200-bar truss

Table 1 Design parameters of the benchmark trusses

	The 10-bar truss	The 37-bar truss	The 72-bar truss	The 52-bar truss	The 120-bar truss	The 200-bar truss
Design variables	$A_i, i = 1,2,\ldots,10$	$A_i, i = 1,2,\ldots,14$; $y_j, j = 3,5,7,9,11$	$G_i, i = 1,2,\ldots,16$	$G_i, i = 1,2,\ldots,8$; z_A, z_B, z_F, x_B, x_F	$G_i, i = 1,2,\ldots,7$	$G_i, i = 1,2,\ldots,29$
Limits $f(Hz)$	$f_1 \geq 7$, $f_2 \geq 15$, $f_3 \geq 20$	$f_1 \geq 20$, $f_2 \geq 40$, $f_3 \geq 60$	$f_1 \geq 4$, $f_3 \geq 6$	$f_1 \leq 15.916$, $f_2 \geq 28.648$	$f_1 \geq 9$, $f_2 \geq 11$	$f_1 \geq 5$, $f_2 \geq 10$, $f_3 \geq 15$
Size variables, $A_i(\text{cm}^2)$	[0.645, 50]	[1, 10]	[0.645, 30]	[1, 10]	[1, 129.3]	[0.1, 30]
Shape variables	–	$y_j \in [0.1, 3]$ m	–	± 2 m	–	–
Material density $\rho(\text{kg/m}^3)$	2770	7800	2770	7800	7971.81	7860
Young modulus $E(Pa)$	6.98×10^{10}	2.1×10^{11}	6.98×10^{10}	2.1×10^{11}	$2.1 \times 10^{11} Pa$	2.1×10^{11}

MSOS algorithm was evaluated for different continuous section truss design. Furthermore, the results are compared with previous results obtained through various existing MHs like FA, TLBO, OC, GA, CSS, CBO, TWO, SOS, and VPS.

5.1 The 10-bar truss

The first benchmark problem of 10-bar truss, shown in Fig. 2, is considered here to investigate the effect of various design parameters (given in Table 1). This problem results already been examined by various researchers is shown in Table 2. For given truss problem, ten continuous design variables are considered for sizing function and at all free nodes (nodes 1–4) of the given truss a 454.0 kg lumped mass is added, as shown in Fig. 2.

MSOS is tested check effects on size optimization by assuming population size as 20 and maximum Function Evaluations as 4000. Table 2 illustrates design variables, truss masses, Standard Deviation (SD) of mass, FE, and frequency responses obtained for 100 independent runs. The result table presents that MSOS and SOS find the minimum mass 524.5747 and 525.2789 kg, respectively. In addition, MSOS finds the best solution as compared to similar results detailed in literature. Therefore, the results of MSOS are compared with the results of the other MHs. The mass saving for MSOS is 18.18, 13.41, 10.57, 7.38, 4.68, 10.42, 6.71, 4.52, 11.16, 7.66, 0.70, 0.35, 0.25, 0.70, and 0.16 kg as compared to those obtained from NHGA, PSO, NHPGA, CSS, enhanced CSS, HS, FA, CSS-BBBC, hybrid OC-GA, TWO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively.

Mean mass (Avg.) obtained using MSOS and SOS is 527.7970 and 531.4033 kg, respectively. As per results, MSOS presents the best Avg. solution among the considered MHs. It is also observed that SOS and its variants find better Avg. solution as compared to similar solutions presented in literature. The Avg. mass saving for MSOS is 13.09, 8.59, 10.73, 9.88, 7.27, 7.75, 3.61, 0.83, 0.75, 0.91, and 2.23 kg as compared to those obtained from PSO, CSS, enhanced CSS, HS, FA, TWO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively.

Table 2 Best solutions obtained using different MHs for the 10-bar truss

Design variable	Wei et al. [54]	Gomes [16]	Wei at al. (2011)	Kaveh and Zolghadr [26]		Miguel and Miguel (2012)		Kaveh and Zolghadr [27]	Zuo et al. [59]	Kaveh and Zolghadr [30]	Tejani et al. [46]				Tejani et al. (2018)	Proposed work
	NHGA	PSO	NHPGA	CSS	enhanced CSS	HS	FA	CSS-BBBC	hybrid OC-GA	TWO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
A_1	42.234	37.712	36.630	38.811	39.569	34.282	36.198	35.274	37.284	34.544	35.3794	34.4523	35.3013	36.4206	35.2654	35.2834
A_2	18.555	9.959	13.043	9.0307	16.740	15.653	14.030	15.463	9.445	15.148	14.8826	14.9767	14.8119	14.3010	14.6803	14.4487
A_3	38.851	40.265	34.229	37.099	34.361	37.641	34.754	32.11	35.051	37.088	35.7321	36.1675	34.9522	34.1835	34.4273	34.5268
A_4	11.222	16.788	15.289	18.479	12.994	16.058	14.900	14.065	19.262	14.813	14.3069	14.6638	14.9436	15.5395	14.9605	14.6773
A_5	4.783	11.576	0.645	4.479	0.645	1.069	0.645	0.645	2.783	0.646	0.6450	0.6680	0.6450	0.6450	0.6450	0.6450
A_6	4.451	3.955	4.8472	4.205	4.802	4.740	4.672	4.880	5.450	4.613	4.7142	4.5484	4.5828	4.6247	4.5927	4.5878
A_7	21.049	25.308	22.140	20.842	26.182	22.505	23.467	24.046	19.041	24.373	24.1569	23.9613	23.5712	22.2793	23.3417	23.5452
A_8	20.949	21.613	27.983	23.023	21.260	24.603	25.508	24.340	27.939	23.72	23.6047	23.4914	23.5602	24.8589	23.8236	24.1081
A_9	10.257	11.576	15.034	13.763	11.766	12.867	12.707	13.343	14.950	12.318	12.1590	12.0449	11.9314	12.9163	12.8497	12.7202
A_{10}	14.342	11.186	10.216	11.414	11.392	12.099	12.351	13.543	10.361	12.618	12.0061	12.4632	13.0401	11.8151	12.5321	12.4136
Mass (kg)	542.75	537.98	535.14	531.95	529.25	534.99	531.28	529.09	535.73	532.23	525.2789	524.9274	524.8289	525.2702	524.7341	524.5747
f_1 (Hz)	7.008	7.000	7.0003	7.000	7.000	7.0028	7.0002	7.000	7.0007	7.0000	7.0005	7.0001	7.0003	7.0007	7.0001	7.0000
f_2 (Hz)	18.148	17.786	16.080	17.442	16.238	16.7429	16.1640	16.119	17.030	16.1599	16.2484	16.2437	16.1997	16.2072	16.1703	16.1666
f_3 (Hz)	20.000	20.000	20.002	20.031	20.000	20.0548	20.0029	20.075	20.156	20.000	19.9999	20.0064	20.0022	19.9996	20.0024	20.0012
FE	–	2000	–	4000	4000	20,000	5000	4000	8000	–	4000	4000	4000	4000	4000	4000
Avg	–	540.89	–	536.39	538.53	537.68	535.07	–	–	535.55	531.4033	528.6291	528.5501	528.7075	530.0286	527.7970
SD	4.864	6.84	–	3.32	5.97	2.49	3.64	–	–	3.24	4.2243	3.4999	2.9827	2.8779	3.4763	2.9121

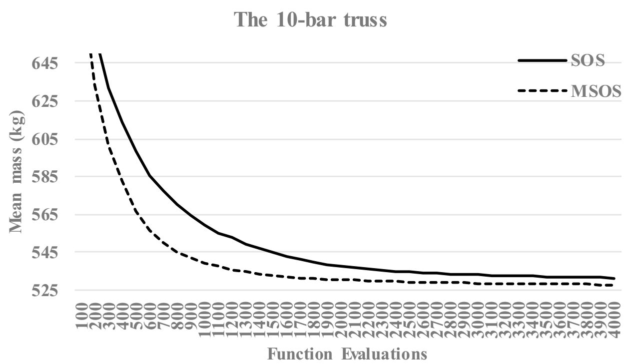

Fig. 8 Convergence curve of the 10-bar truss

MSOS and SOS find SD as 2.9121 and 4.2243 kg, respectively. It is evident that MSOS finds better solution as SD compare to SOS and other MHs except HS and SOS-ABF1&2. It should be noted that maximum *FE* used in MSOS and SOS is fairly small as compared to HS, FA, and hybrid OC-GA. This valuation specifies that the solutions obtained using MSOS and SOS are more reliable and superior with the other solutions stated in literature. However, few studies are available in the literature which demonstrate competitive edge to proposed algorithm like VPS which gives Avg. and SD values of 535.64 and 2.55 kg, respectively, after 4620 analyses, a novel adaptive hybrid evolutionary firefly algorithm (AHEFA) suggested by Lieu et al. [88] which gives Avg. and SD values of 525.1623 and 1.9155 kg, respectively, after 5860 function analyses, Roulette wheel selection-Elitist-Differential Evolution (ReDE) recommended by Ho-Huu et al. [89] results into Avg. and SD values of 525.7039 and 1.3415 kg, respectively, after 20,000 average function analyses. Kaveh and Javadi [90] suggested an efficient hybrid algorithm HRPSO (which compound particle swarm, ray optimizer, and harmony search strategy) which gives Avg. and SD values of 524.88 and 2.253 kg, respectively, after 10 independent runs. Similarly, cyclical parthenogenesis algorithm (CPA) proposed by Kaveh and Zolghadr (2016) gives Avg. and SD values of 533.66 and 2.67 kg, respectively, after 12,800 structural analyses.

Figure 8 shows a convergence graph of mean mass with respect to FE for MSOS and SOS. The mean mass is computed by considering the average mass of all runs for each generation. The graph indicates that MSOS converges faster in nearly 2000 FE. In addition, it is identified that MSOS outperforms the basic SOS.

5.2 The 37-bar truss

The second benchmark problem considered here is shown in Fig. 3, which is a 37-bar truss, simply supported bridge. Initially it was considered by Wang et al. [53] and later many researchers investigate this truss problem (as shown in Table 3). Various design parameters considered here is depicted in Table 1. In addition, all free nodes here considered having a lumped mass of 10 kg attached to the lower chord of truss. Moreover, lower cords of given truss problem are assumed to have fixed rectangular cross-section of 0.4 cm^2, whereas the remaining bars are clustered into fourteen groups by considering symmetry of structure about its middle vertical plane. Here lower nodes are considered to be fixed and upper nodes have a possibility of shifting vertically as structural symmetry has been considered along vertical plane. Due to above consideration this problem has five shape and fourteen size variables.

MSOS and SOS are investigated by assuming population size as 20 and *FE* as 4000. The solutions are presented in Table 3. MSOS and SOS find the best solution as 360.3018 and 360.8658 kg, respectively. The results show that SOS and its variants find nearly similar and better solutions with similar results reported in literature. However, maximum FE used in SOS and its variants is very small as compared to PSO, HS, and FA.

MSOS and SOS present Avg. solutions as 362.9610 and 364.8521 kg, respectively. MSOS and SOS present SD of mass as 1.7265 and 4.2278 kg, respectively. It observed from the solutions that that MSOS archives better solution as Avg. and SD of mass among the considered MHs for 4000 FE. Meanwhile, many competitive solutions are also available in literature like a new Particle Swarm Ray Optimization (PSRO) proposed by Kaveh and Zolghadr [92] which gives the result of mean weight and standard deviation after 20 independent runs as 362.65 and 1.30 kg, respectively. Similarly, CPA gives Avg. and SD values of 360.93 and 0.65 kg, respectively, after 12,800 structural analyses, VPS gives Avg. and SD values of 360.23 and 0.22 kg, respectively, after 7940 structural analyses. In addition, CBO gives the solution of Avg. 360.4463 kg and SD of 0.35655 kg after 6000 number of analysis. Moreover, ReDE results into Avg. and SD values of 359.9944 and 0.1493 kg, respectively, after 12,579 average function analyses.

Figure 9 shows the convergence graph of the 37-bar truss. Graph is plotted between mean mass vs. FE for the proposed algorithms. The convergence graph indicates that MSOS converge faster and achieves good optimal results as compared to SOS. Moreover, MSOS shows early convergence nearly within 3000 FE. In addition, it is identified that from the obtained results, MSOS performs better compare to basic SOS. Therefore, the performance of the proposed modified algorithm is better as compared to its basic version in terms of statistical results and convergence.

Table 3 Best solutions obtained using different MHs for the 37-bar truss

Design variable	Wang et al. [53]	Wei et al. (2005)	Gomes [16]	Wei at al. (2011)	Kaveh and Zolghadr [26]		Miguel and Miguel (2012)		Kaveh and Zolghadr [30]	Tejani et al. [46]				Tejani et al. [45]	Proposed work
	OC	GA	PSO	NHPGA	CSS	enhanced CSS	HS	FA	TWO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
y_3, y_{19}	1.2086	1.1998	0.9637	1.09693	0.8726	1.0289	0.8415	0.9392	1.0039	0.9598	0.9168	0.9413	0.9060	0.9257	1.0111
y_5, y_{17}	1.5788	1.6553	1.3978	1.45558	1.2129	1.3868	1.2409	1.3270	1.3531	1.3867	1.2980	1.3393	1.2665	1.3188	1.4030
y_7, y_{15}	1.6719	1.9652	1.5929	1.59539	1.3826	1.5893	1.4464	1.5063	1.5339	1.5698	1.4777	1.5434	1.4834	1.4274	1.6095
y_9, y_{13}	1.7703	2.0737	1.8812	1.76551	1.4706	1.6405	1.5334	1.6086	1.6768	1.6687	1.6046	1.6744	1.6004	1.5806	1.7610
y_{11}	1.8502	2.3050	2.0856	1.87413	1.5683	1.6835	1.5971	1.6679	1.7728	1.7203	1.6596	1.7571	1.6397	1.6548	1.8513
A_1, A_{27}	3.2508	2.8932	2.6797	2.62463	2.9082	3.4484	3.2031	2.9838	2.8892	2.9038	2.8448	2.9344	3.3609	2.6549	2.9619
A_2, A_{26}	1.2364	1.1201	1.1568	1.0000	1.0212	1.5045	1.1107	1.1098	1.0949	1.0163	1.0785	1.0256	1.0203	1.0383	1.0202
A_3, A_{24}	1.0000	1.0000	2.3476	1.00176	1.0363	1.0039	1.1871	1.0091	1.0213	1.0033	1.0000	1.0095	1.0169	1.0000	1.0000
A_4, A_{25}	2.5386	1.8655	1.7182	2.07586	3.9147	2.5533	3.3281	2.5955	2.6776	3.1940	2.8906	2.5838	2.6772	3.0083	2.3282
A_5, A_{23}	1.3714	1.5962	1.2751	1.22071	1.0025	1.0868	1.4057	1.2610	1.1981	1.0109	1.0335	1.1569	1.0166	1.0024	1.1719
A_6, A_{21}	1.3681	1.2642	1.4819	1.48922	1.2167	1.3382	1.0883	1.1975	1.1387	1.5877	1.2119	1.2548	1.2244	1.4499	1.2374
A_7, A_{22}	2.4290	1.8254	4.6850	2.30847	2.7146	3.1626	2.1881	2.4264	2.6537	2.4104	3.1886	2.5104	2.7056	3.1724	2.1430
A_8, A_{20}	1.6522	2.0009	1.1246	1.43236	1.2663	2.2664	1.2223	1.3588	1.4171	1.3864	1.3435	1.4626	1.5535	1.2661	1.5308
A_9, A_{18}	1.8257	1.9526	2.1214	1.64678	1.8006	1.2668	1.7033	1.4771	1.3934	1.6276	1.7247	1.5245	1.4833	1.4659	1.4839
A_{10}, A_{27}	2.3022	1.9705	3.8600	2.87072	4.0274	1.7518	3.1885	2.5648	2.7741	2.3594	2.6980	2.4586	2.4032	2.9013	2.4001
A_{11}, A_{17}	1.3103	1.8294	2.9817	1.50405	1.3364	2.7789	1.0100	1.1295	1.2759	1.0293	1.1401	1.1888	1.0000	1.1537	1.1678
A_{12}, A_{15}	1.4067	1.2358	1.2021	1.31328	1.0548	1.4209	1.4074	1.3199	1.2776	1.3721	1.2840	1.3765	1.4982	1.3465	1.5085
A_{13}, A_{16}	2.1896	1.4049	1.2563	2.32277	2.8116	1.0100	2.8499	2.9217	2.1666	2.0673	2.3044	2.2341	2.7480	2.6850	2.0768
A_{14}	1.0000	1.0000	3.3276	1.04258	1.1702	2.2919	1.0269	1.0004	1.0099	1.0000	1.0000	1.0007	1.0072	1.0000	1.0075
Mass (kg)	366.50	368.84	377.20	363.032	362.84	362.38	361.50	360.05	360.27	360.8658	360.4260	359.9050	360.5007	360.7432	360.3018
f_1 (Hz)	20.0850	20.0013	20.0001	20.0819	20.0000	20.0028	20.0037	20.0024	20.0279	20.0366	20.0230	20.0052	20.0023	20.0119	20.0017
f_2 (Hz)	42.0743	40.0305	40.0003	40.0961	40.0693	40.0155	40.0050	40.0019	40.0146	40.0007	40.0394	40.0048	40.0363	40.0964	40.0018
f_3 (Hz)	62.9383	60.0000	60.0001	60.0321	60.6982	61.2798	60.0082	60.0043	60.0946	60.0138	60.0339	60.0077	60.0065	60.0066	60.0164
FE	–	–	12,500	–	4000	4000	20,000	5000	–	4000	4000	4000	4000	4000	4000
Avg	–	–	381.2	–	366.77	365.75	362.04	360.37	363.75	364.8521	363.3662	363.0816	363.6336	363.3978	362.9610
SD	–	9.0325	4.26	–	3.742	3.461	0.52	0.26	2.48	2.9650	2.1704	1.8304	2.0771	1.5675	1.7265

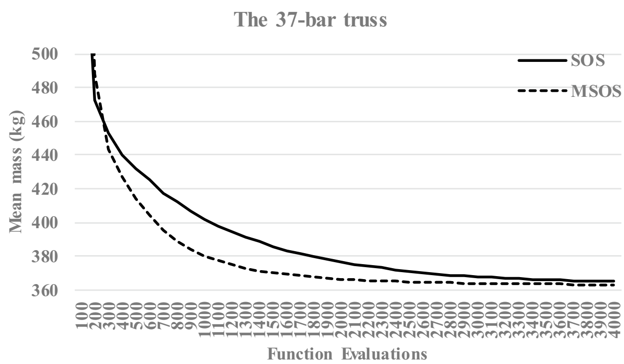

Fig. 9 Convergence curve of the 37-bar truss

5.3 The 72-bar truss

The third benchmark truss problem taken here is manifested in Fig. 4. Similar to earlier cases this truss problem also has been investigated by many researchers at large scale as sizing problem. Table 1 represents the deign consideration for this problem. Here due to assumption of vertical plane symmetry of structure similar to previous cases, sixteen groups of bars are considered. Furthermore, at all top position nodes (nodes 1–4) a lumped mass of 2770 kg is assumed to be attached, as shown in Fig. 4.

MSOS and SOS are investigated by assuming population size as 20 and *FE* as 4000. The solutions are presented in Table 4. The best solution found using MSOS and SOS is 324.346 and 325.5585 kg, respectively. It is observed from solutions that MSOS performs better compared to SOS and its other variants and similar solutions reported in literature. Moreover, MSOS performs better among considered MHs. Therefore, results obtained using MSOS are compared with the results of the other MHs mass saving for MSOS is 4.47, 4.05, 3.16, 0.41, 3.22, 3.23, 4.48, 1.21, 0.74, 0.34, 0.89, and 0.72 kg compared to those obtained from CSS, enhanced CSS, CSS-BBBC, CBO, TLBO, MC-TLBO, TWO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively.

The results signify that MSOS and SOS find Avg. solution as 326.7847 and 331.1228 kg, respectively. MSOS and SOS find SD of mass as 2.1968 and 4.2278 kg, respectively. It can be seen from the results that MSOS finds best result as Avg. and SD of solutions among the considered MHs with 4000 FEs. It observed that maximum FE consumed by CBO, TLBO, and MC-TLBO is significantly better as compared to the other MHs. However, few studies are available in the literature which illustrate competitive solutions than proposed algorithm like CPA gives Avg. and SD values of 330.91 and 1.84 kg, respectively, after 12,800 structural analyses, VPS gives Avg. and SD values of 327.670 and 0.018 kg, respectively, after 4720 structural analyses, HRPSO gives Avg. and SD values of 324.497(in lb.) and 3.948 kg, respectively, after 10 independent runs. Similarly, ReDE results into Avg. and SD values of 324.3219 and 0.0516 kg, respectively, after 11,116 average function analyses, AHEFA gives Avg. and SD values of 324.4109 and 0.2420 kg, respectively, after 8860 function analyses.

Figure 10 illustrates a convergence graph of the mean mass for MSOS and SOS. As observed from the graph, the mean mass for 4000 FE is 326.7847 kg and 331.1228 kg for MSOS and SOS, respectively. It can be seen from the convergence graph that MSOS converges better than SOS. This study specifies that the solutions obtained using MSOS are more reliable and proficient as compared to similar solutions obtained using other MHs.

5.4 The 52-bar truss

The fourth problem selected, as shown in Fig. 5, is 52-bar dome shape truss problem. This truss problem was first studied by Lin et al. (1982) for shape and sizing optimization and later it was considered by several other scholars, as shown in Table 5. For this problem various design parameters consideration are illustrated in Table 1. As in first case, here a 50 kg lumped mass is assumed to be attached at free nodes. The elements of structure are clustered into eight groups considering *z*-axis symmetry and also to keep dome symmetric free nodes are allowed to shift ± 2 m in each direction of the vertical plane.

MSOS and SOS are investigated by assuming population size as 20 and FE as 4000. Table 5 illustrates the solutions obtained using MSOS, SOS, and other MHs stated in literature. The solutions show that MSOS and SOS propose trusses with the best mass of 193.773 and 195.4969 kg, respectively. It is identified that MSOS ranks first, whereas ISOS ranks second among the considered MHs. The mass saving for MSOS is 104.23, 42.27, 34.61, 11.46, 3.56, 21.17, 3.76, 3.54, 1.58, 0.48, 1.72, 1.04, 1.40, 4.49, and 0.98 kg compared to those obtained from Bi-factor algorithm, NGHA, PSO, CSS, enhanced CSS, HS, FA, CSS-BBBC, DPSO, TWO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively.

The results signify that MSOS and SOS present Avg. solutions as 204.4279 and 214.6676 kg, respectively. The results indicate that MSOS finds better Avg. solutions as compared to other MHs stated in literature except DPSO. In addition, the maximum FE consumed in the proposed MHs is fairly small as compared to PSO, HS, FA, and DPSO. MSOS and SOS find SD as 8.4014 and 15.1499 kg, respectively. However, few studies are also available in the literature which exhibit competitive solutions than proposed algorithm like CPA gives Avg. and SD values of 198.81 and 3.71 kg, respectively, after 12,800 structural analyses, HRPSO gives Avg. and SD values of 193.361 and 17.637 kg,

Table 4 Best solutions obtained using different MHs for the 72-bar truss

Design variable	Kaveh and Zolghadr [26]		Kaveh and Zolghadr [27]	Kaveh and Mahdavi [23]	Farshchin et al. [12]		Kaveh and Zolghadr [30]	Tejani et al. [46]				Tejani et al. (2018)	Proposed work
	CSS	enhanced CSS	CSS-BBBC	CBO	TLBO	MC-TLBO	TWO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
A_1–A_4	2.528	2.522	2.854	3.3699	3.5491	3.4188	3.380	3.6957	4.1820	3.6273	3.8745	3.3563	3.3177
A_5–A_{12}	8.704	9.109	8.301	7.3428	7.9676	7.9263	8.086	7.1779	7.8990	7.9416	7.6185	7.8726	8.0232
A_{13}–A_{16}	0.645	0.648	0.645	0.6468	0.6450	0.6450	0.647	0.6450	0.6450	0.6460	0.6450	0.6450	0.6458
A_{17}–A_{18}	0.645	0.645	0.645	0.6457	0.6450	0.6450	0.646	0.6569	0.6450	0.6450	0.6957	0.6450	0.6450
A_{19}–A_{22}	8.283	7.946	8.202	8.0056	8.1532	8.0143	8.89	7.7017	8.0149	7.5653	8.4112	8.5798	8.1354
A_{23}–A_{30}	7.888	7.703	7.043	8.0185	7.9667	7.9603	8.136	7.9509	8.1772	8.0171	7.7833	7.6566	7.8910
A_{31}–A_{34}	0.645	0.647	0.645	0.6458	0.6450	0.6450	0.654	0.6450	0.6450	0.6714	0.6450	0.7417	0.6535
A_{35}–A_{36}	0.645	0.6456	0.645	0.6457	0.6450	0.6450	0.647	0.6450	0.6450	0.6450	0.6450	0.6450	0.6450
A_{37}–A_{40}	14.666	13.465	16.328	12.4585	12.9272	12.7903	13.097	12.3994	12.4516	13.4781	12.0976	13.0864	12.3358
A_{41}–A_{48}	6.793	8.250	8.299	8.1211	8.1226	8.1013	8.101	8.6121	7.7290	7.6531	7.7086	8.0764	7.9314
A_{49}–A_{52}	0.645	0.645	0.645	0.6460	0.6452	0.6450	0.663	0.6450	0.6525	0.6450	0.6450	0.6450	0.6450
A_{53}–A_{54}	0.645	0.646	0.645	0.6459	0.6450	0.6473	0.646	0.6450	0.6450	0.6450	0.6450	0.6937	0.6454
A_{55}–A_{58}	16.464	18.368	15.048	17.3636	17.0524	17.4615	16.483	17.4827	16.8203	16.6583	16.9516	16.2517	17.4473
A_{59}–A_{66}	8.809	7.053	8.268	8.3371	8.0618	8.1304	7.873	8.1502	7.9846	8.1609	8.7289	8.1703	7.9102
A_{67}–A_{70}	0.645	0.645	0.645	0.6460	0.6450	0.6450	0.651	0.6740	0.6742	0.6450	0.6450	0.6450	0.6452
A_{71}–A_{72}	0.645	0.646	0.645	0.6476	0.6450	0.6451	0.657	0.6550	0.6450	0.6450	0.6450	0.6450	0.6450
Mass (kg)	328.814	328.393	327.507	324.7552	327.568	327.5750	328.83	325.5585	325.086	324.6897	325.2317	325.0682	324.346
f_1 (Hz)	4.000	4.000	4.000	4.0000	4.000	4.000	4.000	4.0023	4.0045	4.0013	4.0016	4.0000	4.001
f_3 (Hz)	6.006	6.004	6.004	6.0000	6.000	6.000	6.000	6.0020	6.0019	6.0002	6.0003	6.0008	6.001
FE	4000	4000	4000	6000	15,000	15,000	–	4000	4000	4000	4000	4000	4000
Avg	337.70	335.77	–	330.4154	328.684	327.6930	336.1	331.1228	328.6582	328.4621	334.9979	329.4699	326.7847
SD	5.42	7.20	–	7.7063	0.73	0.1250	5.8	4.2278	2.7948	2.4600	6.0566	2.6642	2.1968

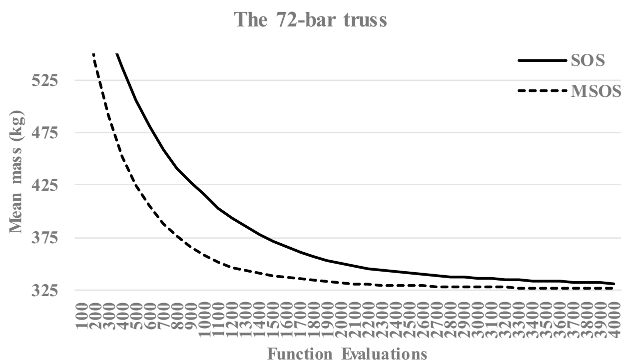

Fig. 10 Convergence curve of the 72-bar truss

respectively, after 10 independent runs, AHEFA gives Avg. and SD values of 198.7290 and 4.4108 kg, respectively, after 12,120 function analyses. Similarly, ReDE results into Avg. and SD values of 195.4260 and 3.8587 kg, respectively, after 14,749 average function analyses. In addition, an Improved ray optimization (IRO) Kaveh et al. [82] gives Avg. and SD values of 196.43 and 1.81 kg, respectively, after 17,000 structural analyses.

It is observed from the solutions that that MSOS gives better Avg. solution and SD as compared to basic SOS. Figure 11 compares the convergence characteristic curve of the mean mass for MSOS and SOS. The convergence graph indicates that MSOS converges faster and achieves good optimal results as compared to SOS. Moreover, MSOS shows early convergence nearly within 3000 FE. This study indicates that the results of MSOS, SOS, and its other variants are more reliable and proficient as compared to the results of the other considered MHs. In addition, MSOS performs more efficiently as compared to basic SOS.

5.5 The 120-bar truss

The fifth benchmarks problem considered here is 120 bar truss problem, as shown in Fig. 6.This truss has a 3-D dome structure which was first considered by Kaveh and Zolghadr [27] for size optimization. Again, for this truss problem the design parameters considered is shown in Table 1. Conversely rather than using single lumped mass as done in previous cases, here a lumped mass of 3000 kg is added at node1, from 2 to 13 nodes 500 kg and for rest of free nodes a 100 kg mass is added. Again, based on the z-axis symmetry assumption the elements are clustered into seven group here.

MSOS and SOS are investigated by assuming population size as 20 and FE as 4000. Table 6 illustrates the obtained solutions using SOS and other MHs. It shows that MSOS and SOS find the truss problems with the best mass of 8708.3180 and 8713.3030 kg, respectively. The results show that SOS, MSOS, and its other variants give better solutions compared to similar solutions detailed in literature. In addition, MSOS ranks first among the considered MHs. MSOS gives mass saving of 496.19, 338.02, 180.81, 463.61, 182.16, 4.99, 3.79, 2.01, 8.63, and 1.74 kg compared to solutions obtained from CSS, CSS-BBBC, CBO, PSO, DPSO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively.

The Avg. solution for MSOS and SOS are 8720.9734 and 8735.3452 kg, respectively. It is observed that MSOS finds better Avg. solution among the considered MHs. The Avg. mass benefit for MSOS is 170.28, 530.87, 175.02, 14.37, 6.45, 4.33, 69.72, and 7.62 kg as compared to those obtained from CBO, PSO, DPSO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS, respectively. It is observed evidently that MSOS gives better Avg. solution as compared to similar solutions presented in literature. However, few studies are available in the literature which demonstrate competitive solutions than proposed algorithm like VPS gives Avg. and SD values of 8896.04 and 6.65 kg, respectively, after 6860 structural analyses. Similarly, ReDE results into Avg. and SD values of 8707.5233 and 0.1543 kg, respectively, after 5380 average function analyses, AHEFA gives Avg. and SD values of 8707.5580 and 0.2535 kg, respectively, after 3560 function analyses, IRO gives Avg. and SD values of 8905.21 and 4.92 kg, respectively, after 16,300 structural analyses. In addition, Jalili and Talatahari (2018) proposed a hybrid Charged System Search with Migration-based Local Search algorithm (CSS-MBLS) which gives Avg. and SD values of 8715.60 and 4.95 kg, respectively, after 2400 structural analyses. Furthermore, CBO gives the solution of Avg. 8,891.2540 and SD of 1.7926 kg after 6000 number of analysis.

MSOS and SOS present SD as 11.7324 and 17.9011, respectively. CBO and DPSO rank first and second, respectively, in terms of SD but maximum FE consumed by these MHs is 50% higher compare to SOS and its variants. MSOS and SOS-ABF2 are nearly similar in SD and perform better compare to others with 4000 *FEs*. Figure 12 displays convergence of the mean solutions obtained for MSOS and SOS. It is observed from the convergence graph of MSOS that the objective function converges rapidly within initial 2500 *FEs*. Convergence graph also indicates that MSOS converges better than SOS and converges to better solutions. This study signifies that the solution obtained using MSOS are more reliable and proficient as compared to SOS and other solutions presented in literature without violation of constraints.

Table 5 Best solutions obtained using different MHs for the 52-bar truss

Design variable	Lin et al. (1982)	Wei et al. (2005)	Gomes [16]	Kaveh and Zolghadr [26]		Miguel and Miguel [36]		Kaveh and Zolghadr [27]	Kaveh and Zolghadr [29]	Kaveh and Zolghadr [30]	Tejani et al. [46]				Tejani et al. (2018)	Proposed work
	Bi-factor algorithm	NGHA	PSO	CSS	Enhanced CSS	HS	FA	CSS-BBBC	DPSO	TWO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
z_A	4.3201	5.8851	5.5344	5.2716	6.1590	4.7374	6.4332	5.331	6.1123	6.012	5.7624	5.9650	6.0120	5.8950	6.1631	5.8551
x_B	1.3153	1.7623	2.0885	1.5909	2.2609	1.5643	2.2208	2.134	2.2343	1.598	2.3239	2.3240	2.4250	2.4237	2.4224	2.3651
z_B	4.1740	4.4091	3.9283	3.7093	3.9154	3.7413	3.9202	3.719	3.8321	4.287	3.7379	3.7002	3.7000	3.7030	3.8086	3.7000
x_F	2.9169	3.4406	4.0255	3.5595	4.0836	3.4882	4.0296	3.935	4.0316	3.641	3.9842	3.9636	4.0201	3.9926	4.1080	3.9960
z_F	3.2676	3.1874	2.4575	2.5757	2.5106	2.6274	2.5200	2.500	2.5036	2.888	2.5121	2.5000	2.5000	2.5000	2.5018	2.5000
A_1–A_4	1.00	1.0000	0.3696	1.0464	1.0335	1.0085	1.0050	1.0000	1.0001	2.1245	1.0988	1.0000	1.0000	1.0000	1.0074	1.0000
A_5–A_8	1.33	2.1417	4.1912	1.7295	1.0960	1.4999	1.3823	1.3056	1.1397	1.1341	1.0031	1.1797	1.0000	1.0000	1.0003	1.0459
A_9–A_{16}	1.58	1.4858	1.5123	1.6507	1.2449	1.3948	1.2295	1.4230	1.2263	1.187	1.1956	1.2109	1.1280	1.0000	1.1982	1.2388
A_{17}–A_{20}	1.00	1.4018	1.5620	1.5059	1.2358	1.3462	1.2662	1.3851	1.3335	1.318	1.4563	1.4800	1.4466	1.5759	1.2787	1.4634
A_{21}–A_{28}	1.71	1.911	1.9154	1.7210	1.4078	1.6776	1.4478	1.4226	1.4161	1.3637	1.3773	1.3977	1.4298	1.4046	1.4421	1.4070
A_{29}–A_{36}	1.54	1.0109	1.1315	1.0020	1.0022	1.3704	1.0000	1.0000	1.0001	1.0299	1.0055	1.0229	1.0032	1.0000	1.0000	1.0028
A_{37}–A_{44}	2.65	1.4693	1.8233	1.7415	1.6024	1.4137	1.5728	1.5562	1.5750	1.3479	1.7397	1.6747	1.7686	1.6494	1.4886	1.5907
A_{45}–A_{52}	2.87	2.1411	1.0904	1.2555	1.4596	1.9378	1.4153	1.4485	1.4357	1.4446	1.3084	1.3033	1.2770	1.5664	1.4990	1.3675
Mass (kg)	298.0	236.046	228.381	205.237	197.337	214.94	197.53	197.309	195.351	194.25	195.4969	194.8089	195.1730	198.2630	194.7483	193.773
f_1 (Hz)	15.22	12.81	12.751	9.246	11.849	12.2222	11.3119	12.987	11.315	9.265	12.7144	11.8992	12.2594	12.8140	12.5459	12.5185
f_2 (Hz)	29.28	28.65	28.649	28.648	28.649	28.6577	28.6529	28.648	28.648	28.667	28.6540	28.6478	28.6576	28.7301	28.6518	28.6546
FE	–	–	11,270	4000	4000	20,000	10,000	4000	6000	–	4000	4000	4000	4000	4000	4000
Avg	-	-	234.3	213.101	205.617	229.88	212.80	-	198.71	214.25	214.6676	210.7033	211.5683	224.5050	207.5498	204.4279
SD	–	37.462	5.22	7.391	6.924	12.44	17.98	–	13.85	12.64	15.1499	11.8339	12.7871	17.8552	8.7354	8.4014

Fig. 11 Convergence curve of the 52-bar truss

5.6 The 200-bar truss

Figure 7 illustrates the sixth benchmark problem, i.e., 200 bar truss optimization problem which is a large-scale size optimization problem. Design parameters like design variables, limits, frequency constraints etc. are presented in Table 1. Here a lumped mass of 100 kg is assumed at all top nodes (nodes 1–5) of structure. Again, based on the *z*-axis symmetry assumption, all the elements are clustered into 29 groups here.

SOS and its variants are assumed with population size of 20 and *FE* of 10,000. Table 7 illustrates the comparative solutions. The best solutions for MSOS and SOS are 2164.47 and 2180.321 kg, respectively. The results show that MSOS gives better solutions compared SOS and its variants and similar results reported in literature. The solutions show that MSOS finds mass saving of 95.39, 134.14, 38.74, 24.61, 15.85, 0.41, 1.33, 43.42, and 4.99 kg as compared to those obtained from CSS, CSS-BBBC, CBO, 2D-CBO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS.

MSOS and SOS find Avg. solutions of 2185.4313 and 2303.3034 kg, respectively. It is observed for the solutions that MSOS gives better Avg. solution than SOS and other solutions. MSOS and SOS find SD as15.4062 and 83.5897, respectively. It can be seen from the solutions that that MSOS finds better result as SD than SOS and other solutions presented in the table except SOS-ABF1. In addition, SOS-ABF1 and MSOS find SD nearly similar. However, few studies are available in the literature which demonstrate competitive edge to proposed algorithm like enhanced colliding bodies' optimization algorithm proposed by Kaveh and Ghazaan [83] gives Avg. and SD values of 2159.93 and 1.57 kg, respectively, after 14,700 number of analyses, CSS-MBLS gives Avg. and SD values of 2,157.40 and 1.04 kg, respectively, after 9600 number of analyses, AHEFA gives Avg. and SD values of 2161.0393 and 0.1783 kg, respectively, after 11,300

Table 6 Best solutions obtained using different MHs for the 120-bar truss

Group no	Kaveh and Zolghadr [27]		Kaveh and Mahdavi [23]	Kaveh and Zolghadr [29]		Tejani et al. [46]				Tejani et al. (2018)	Proposed work
	CSS	CSS-BBBC	CBO	PSO	DPSO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
G_1	21.710	17.478	19.6917	23.494	19.607	19.5203	19.5449	19.5715	19.3806	19.6662	19.5297
G_2	40.862	49.076	41.1421	32.976	41.290	40.8482	40.9483	39.8327	40.4230	39.8539	40.2338
G_3	9.048	12.365	11.1550	11.492	11.136	10.3225	10.4482	10.5879	11.1095	10.6127	10.6856
G_4	19.673	21.979	21.3207	24.839	21.025	20.9277	21.0465	21.2194	21.2086	21.2901	21.2363
G_5	8.336	11.190	9.8330	9.964	10.060	9.6554	9.5043	10.0571	9.9200	9.7911	9.8421
G_6	16.120	12.590	12.8520	12.039	12.758	12.1127	11.9362	11.8322	11.3161	11.7899	11.7177
G_7	18.976	13.585	15.1602	14.249	15.414	15.0313	14.9424	14.7503	14.7820	14.7437	14.7313
Mass (kg)	9204.51	9046.34	8889.1303	9171.93	8890.48	8713.3030	8712.1100	8710.3300	8716.9470	8710.0620	8708.3180
f_1 (Hz)	9.002	9.000	9.0000	9.0000	9.0001	9.0009	9.0011	9.0006	9.0012	9.0001	9.0007
f_2 (Hz)	11.002	11.007	11.0000	11.0000	11.0007	11.0005	11.0003	11.0002	11.0023	10.9998	11.0000
FE	4000	4000	6000	6000	6000	4000	4000	4000	4000	4000	4000
Avg	–	–	8891.2540	9251.84	8895.99	8735.3452	8727.4267	8725.3075	8790.6961	8728.5951	8720.9734
SD	–	–	1.7926	89.38	4.26	17.9011	16.5503	10.6402	55.7294	14.2296	11.7324

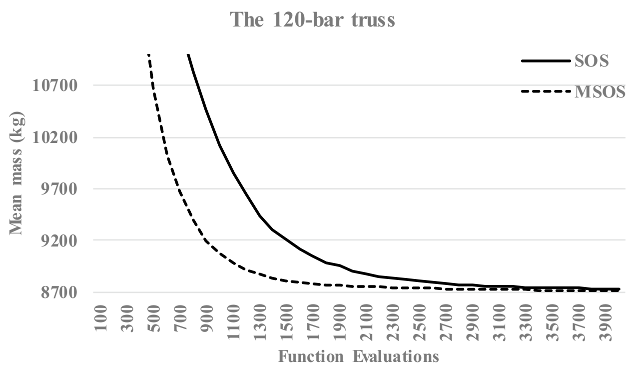

Fig. 12 Convergence curve of the 120-bar truss

function evaluation analyses, MC-TLBO gives Avg. and SD values of 2157.447 and 0.528 kg, respectively, after 23,000 number of analyses.

This test clarifies that the solutions of MSOS are more reliable and proficient as compared to SOS and other solutions reported in literature. Figure 13 shows a convergence graph of the mean mass for MSOS and SOS of the 200-bar truss. It can be seen easily that the graph converges faster for initial 7000 *FEs*. The convergence graph and statistical results indicate that MSOS converges faster and set superior results.

The aforementioned discussion demonstrates the comparison between best mass, average mass, and SD values obtained using the proposed MSOS and SOS algorithms for six benchmark problems. In the tables, the average values exemplify the convergence rate of the algorithm, whereas the SD determines the search consistency. It can be seen from the result summary that performance of SOS has been improved by the proposed modifications. In addition, convergence plots represented above for benchmarks problems with proposed modification manifest the solution with faster convergence with minimum time. The overall performance of MSOS is the significant among the measured MHs. In addition, this study illustrates that the proposed modification outperformance with respect to the basic SOS algorithm. Obtained solutions confirm the merits of the proposed MH.

6 Corroborating MSOS performance with various benchmark functions

This section the twenty-two benchmark functions are extracted from CEC 2014 Special Session and Competition on Real-Parameter Numerical Optimization (Liang et al. 2013) are exploited to manifest the effectiveness of the proposed algorithm. These twenty-two standard problems are epitomized in Table 8. Furthermore, these benchmark problems are subdivided into three segments: Three unimodal Functions (F1–F3), thirteen simple multimodal Functions (F4–F16) and also six hybrid Functions (F17–F22). Moreover, eight distinct optimization algorithms (viz. WWO, BA, HuS, GSA, BBO, IWO, SOS, and MSOS) has been applied to these standard problems and compared to verify the results. In this study, 30-dimensional functions are used with search ranges as [− 100, 100]. Population size is considered as 50 and FE_{max} are taken as 150,000 for proposed algorithm. All results are collected from 60 independent runs on each test function.

Comparative Avg. and SD of fitness values over the 60 runs are presented in Tables 9 and 10, respectively. Statistical tests are essential to check significance improvements by a proposed method over existing methods. Thus, the Friedman rank test on the results of MSOS, SOS, and other state-of-the-art algorithms. The test is performed on the Avg. and SD of functional values obtained. The tables also present the rank sum of the algorithms over the test functions. The results signify that MSOS outperforms other optimizers for unimodal functions followed by SOS and WWO. For multimodal and hybrid functions WWO gives best results followed by IWO and MSOS among the considered algorithms. Moreover, MSOS ranks better compared to SOS for unimodal, multimodal, and hybrid functions. The overall performance of MSOS is second best among the considered algorithms, whereas WWO performs the best on the benchmark functions of unimodal, multimodal, and hybrid functions. These results confirm the merits of the proposed algorithms once more.

7 Conclusion and future perspectives

A modified SOS is presented here for determining the minimum mass design of truss structure subjected to multiple natural frequency constraints with optimal nodal positions and element cross section areas. The proposed algorithm is applied successfully on five benchmark problems of simultaneous size and shape, optimization to investigate their performance. Complementarily, three unimodal functions, thirteen multimodal functions, and six hybrid functions of the CEC2014 are also investigated. A modified parasite vector is proposed here for improvement in exploitation capability of parasitism phase in basic SOS algorithm. Furthermore, an adaptive factor is introduced in basic SOS algorithm to improve its efficiency for complex structures. Motive behind proposed study is to maintain harmony between exploration and exploitation potency of optimization algorithm in search space. Here all benchmark problems are examined by considering constraints like natural

Table 7 Best solutions obtained using different MHs for the 200-bar truss

Group no	Bars	Kaveh and Zolghadr [27]		Kaveh and Mahdavi [24]		Tejani et al. [46]				Tejani et al. [45]	Proposed work
		CSS	CSS-BBBC	CBO	2D-CBO	SOS	SOS-ABF1	SOS-ABF2	SOS-ABF1&2	ISOS	MSOS
G_1	1,2,3,4	1.2439	0.2934	0.3268	0.4460	0.4781	0.2822	0.3058	0.3845	0.3072	0.3951
G_2	5,8,11,14,17	1.1438	0.5561	0.4502	0.4556	0.4481	0.5014	0.5196	0.8524	0.5075	0.4399
G_3	19,20,21,22,23,24	0.3769	0.2952	0.1000	0.1519	0.1049	0.1071	0.1000	0.1130	0.1001	0.1000
G_4	18,25,56,63,94,101,132,139,170,177	0.1494	0.1970	0.1000	0.1000	0.1045	0.1002	0.1092	0.1000	0.1000	0.1000
G_5	26,29,32,35,38	0.4835	0.8340	0.7125	0.4723	0.4875	0.5277	0.5238	0.5084	0.5893	0.4422
G_6	6,7,9,10,12,13,15,16,27,28,30,31,33,34,36,37	0.8103	0.6455	0.8029	0.7543	0.9353	0.8248	0.7956	0.8885	0.8328	0.8130
G_7	39,40,41,42	0.4364	0.1770	0.1028	0.1024	0.1200	0.1300	0.1003	0.1000	0.1431	0.1000
G_8	43,46,49,52,55	1.4554	1.4796	1.4877	1.4924	1.3236	1.4016	1.3119	1.2170	1.3600	1.4808
G_9	57,58,59,60,61,62	1.0103	0.4497	0.1000	0.1000	0.1015	0.1000	0.1056	0.1356	0.1039	0.1000
G_{10}	64,67,70,73,76	2.1382	1.4556	1.0998	1.6060	1.4827	1.4657	1.6178	1.5477	1.5114	1.4549
G_{11}	44,45,47,48,50,51,53,54,65,66,68,69,71,72,74,75	0.8583	1.2238	0.8766	1.2098	1.1384	1.1327	1.1954	1.0568	1.3568	1.2452
G_{12}	77,78,79,80	1.2718	0.2739	0.1229	0.1061	0.1020	0.1196	0.1615	0.4552	0.1024	0.1397
G_{13}	81,84,87,90,93	3.0807	1.9174	2.9058	3.0909	2.9943	3.0262	2.9102	3.4433	2.9024	3.0458
G_{14}	95,96,97,98,99,100	0.2677	0.1170	0.1000	0.7916	0.1562	0.2527	0.1134	0.1000	0.1000	0.1000
G_{15}	102,105,108,111,114	4.2403	3.5535	3.9952	3.6095	3.4330	3.3267	3.5156	3.6060	3.4120	3.3282
G_{16}	82,83,85,86,88,89,91,92,103,104,106,107,109,110, 112,113	2.0098	1.3360	1.7175	1.4999	1.6816	1.5963	1.6227	1.4460	1.4819	1.5484
G_{17}	115,116,117,118	1.5956	0.6289	0.1000	0.1000	0.1026	0.2417	0.3687	0.1893	0.2587	0.3122
G_{18}	119,122,125,128,131	6.2338	4.8335	5.9423	5.2951	5.0739	4.8557	4.6196	5.1791	4.8291	5.1540
G_{19}	133,134,135,136,137,138	2.5793	0.6062	0.1102	0.1000	0.1068	0.1001	0.1543	0.2666	0.1499	0.1032
G_{20}	140,143,146,149,152	3.0520	5.4393	5.8959	4.5288	6.0176	5.4975	5.6545	5.8750	5.5090	5.3092
G_{21}	120,121,123,124,126,127,129,130,141,142,144,145, 147,148,150,151	1.8121	1.8435	2.1858	2.2178	2.0340	2.0829	2.2106	2.5624	2.2221	2.1574
G_{22}	153,154,155,156	1.2986	0.8955	0.5249	0.7571	0.6595	0.8522	0.6688	0.7535	0.6113	0.7208
G_{23}	157,160,163,166,169	5.8810	8.1759	7.2676	7.7999	6.9003	7.5480	7.4241	7.9706	7.3398	7.8337
G_{24}	171,172,173,174,175,176	0.2324	0.3209	0.1278	0.3506	0.2020	0.1279	0.1187	0.3324	0.1559	0.3241
G_{25}	178,181,184,187,190	7.7536	10.98	7.8865	7.8943	6.8356	7.6278	7.5955	7.3386	8.6301	7.9443
G_{26}	158,159,161,162,164,165,167,168,179,180,182,183, 185,186,188,189	2.6871	2.9489	2.8407	2.8097	2.6644	3.0233	2.7572	3.0958	2.8245	2.9322
G_{27}	191,192,193,194	12.5094	10.5243	11.7849	10.4220	12.1430	10.3024	11.1467	9.1512	10.8563	10.3005
G_{28}	195,197,198,200	29.5704	20.4271	22.7014	21.2576	22.2484	21.4034	21.4328	20.7230	20.9142	20.7009
G_{29}	196,199	8.2910	19.0983	7.8840	11.9061	8.9378	10.4810	9.8690	12.1258	10.5305	11.2426
Mass (kg)		2259.86	2298.61	2203.212	2189.08	2180.3210	2164.8840	2165.8010	2207.8880	2169.4590	2164.47
f_1 (Hz)		5.000	5.010	5.0010	5.0016	5.0001	5.0001	5.0000	5.0000	5.0000	5.0000
f_2 (Hz)		15.961	12.911	12.5247	13.3868	13.4306	12.1388	12.3327	13.3064	12.4477	12.8238
f_3 (Hz)		16.407	15.416	15.1845	15.1981	15.2645	15.1284	15.1454	15.5397	15.2332	15.0104
FE		10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
Avg		–	–	2481.492	2308.443	2303.3034	2186.5744	2187.2517	2405.3479	2244.6372	2185.4313
SD		–	–	250.8259	132.5148	83.5897	15.2711	16.9436	128.1578	43.4808	15.4062

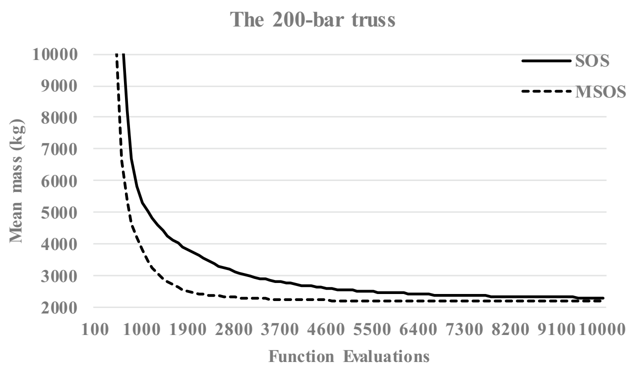

Fig. 13 Convergence curve of the 200-bar truss

frequency, nodal coordinates, and element section area for evaluation of feasibility and effectiveness of the algorithm. Usually design variable like nodal coordinates and element section area are of diverse nature and on collaboration they show complexity and diversified characteristics. In addition, frequency bounds with multiple design variable results into non-linearity and non-convexity of optimization problem.

In this study, the performance of the MSOS algorithm is compared with various existing MHs like NHGA, PSO, NHPGA, CSS, enhanced CSS, HS, FA, CSS-BBBC, hybrid OC-GA, TWO, SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2, and ISOS. Results shows the dominance of MSOS with respect to existing algorithms in parameters like optimum mass, Average mass, and SD of mass. Results also manifest that the combined utilization of adaptive factor and modification in parasitism vector in the proposed algorithm, improves the efficiency of search with good balance between exploration and exploitation potential. In addition, results show that the propose algorithm has high exploration potential during initial function evaluations and also high exploitation capability during remaining function evaluation, which shows its outstanding possession of global exploration and exploitation capacity in search space. To evaluate the performance of the proposed algorithms in benchmark functions, the results of SOS and MSOS are compared with the results of the WWO, BA, HuS, GSA, BBO, and IWO algorithms for the twenty-two benchmark functions proposed in the CEC2014 competition. Overall, MSOS has a better or competitive for obtaining results based on the average and SD of functional values obtained over the stated runs as compared to SOS.

A possible direction for future work would be to extend the proposed method to investigate the multi-objective truss structure design problems, where other objective functions such as joint cost and total construction cost can be taken into account. Application of efficient analysis techniques can facilitate the optimization problem of large-scale structures. It seems that introducing new mathematical and MHs as well as hybridizing and improving the existing ones to address structural optimization with frequency constraints will continue to grow as an active research topic. Moreover, utilization of novel algebraic and graph theoretical methods to decrease the computational time for optimization of different types of regular and near-regular structures can also be a very interesting field for future research. Considering frequency constraints along with stress, displacement, and other types of constraints can also receive more attraction. Inquisitive researchers can analyse this algorithm for various optimization problems, from small-scale engineering design problems to large-scale truss structure design

Table 8 CEC 2014 22 benchmark functions

Test function	Optimum	Test function	Optimum
F1: rotated high conditioned Elliptic function	100	F12: Shifted and Rotated Katsuura Function	1200
F2: rotated bent Cigar function	200	F13: Shifted and Rotated HappyCat Function	1300
F3: rotated discus function	300	F14: Shifted and Rotated HGBat Function	1400
F4: shifted and rotated Rosenbrock's function	400	F15: Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
F5: Shifted and Rotated Ackley's Function	500	F16: Shifted and Rotated Expanded Scaffer's F6 Function	1600
F6: Shifted and Rotated Weierstrass Function	600	F17: Hybrid Function 1 ((F9, F8,F1)	1700
F7: Shifted and Rotated Griewank's Function	700	F18: Hybrid Function 2 (F2, F12, F8)	1800
F8: Shifted Rastrigin's Function	800	F19: Hybrid Function 3 (F7, F6, F4, F14)	1900
F9: Shifted and Rotated Rastrigin's Function	900	F20: Hybrid Function 4 (F12, F3, F13, F8)	2000
F10: Shifted Schwefel's Function	1000	F21: Hybrid Function 5 (F14, F12, F4, F9, F1)	2100
F11: Shifted and Rotated Schwefel's Function	1100	F22: Hybrid Function 6 (F10, F11, F13, F9, F5)	2200

Table 9 Comparative Avg. of fitness values of the CEC2014 (The results of first six algorithms are as per Zheng [76])

Function	WWO	BA	Hus	GSA	BBO	IWO	SOS	MSOS
F1	628064.7331	316593399.3261	5555804.7723	14413625.1299	27262607.1812	1463430.6695	1026753.8150	467711.3550
F2	330.4397	25714756385.1935	10068.1285	8771.2239	4012004.0764	17672.1722	213.1503	231.0785
F3	526.8209	72001.7161	502.0203	45384.2492	13100.3120	8167.4657	938.9390	496.9134
Friedman value of F1-F3	8	24	12	17	20	15	8	4
Friedman rank of F1-F3	2	8	4	6	7	5	2	1
F4	417.0105	3697.5439	506.9362	676.4360	538.7936	500.3255	468.2918	466.2221
F5	519.9999	520.9716	520.7029	519.9990	520.1556	520.0140	520.5639	520.0303
F6	605.9873	636.3693	623.0650	619.5872	613.9623	602.2138	610.8746	611.5147
F7	700.0037	910.6678	700.0407	700.0001	701.0283	700.0337	700.0161	700.0216
F8	801.1436	1070.3076	940.1063	800.4991	877.4573	843.7475	852.1217	824.6915
F9	961.0930	1250.0944	1011.9988	1059.7399	951.4286	946.0714	970.5093	966.3686
F10	1581.5778	6426.1095	2253.5001	4392.2443	1002.1744	2565.2591	2107.2343	1378.4436
F11	3349.4633	8152.1644	3302.9108	5099.2681	3247.3542	2887.3064	4017.4845	3636.8347
F12	1200.0995	1202.5771	1200.1870	1200.0011	1200.2257	1200.0355	1200.6611	1200.3483
F13	1300.2617	1304.0199	1300.3921	1300.2972	1300.5091	1300.2789	1300.4233	1300.3700
F14	1400.2169	1473.1361	1400.2377	1400.2540	1400.4439	1400.2360	1400.3309	1400.2643
F15	1503.2828	194533.2621	1517.0308	1503.2887	1514.6242	1503.6932	1517.6988	1510.9396
F16	1610.4351	1612.9981	1611.7074	1613.6691	1609.9125	1610.4324	1610.6564	1610.4357
Friedman value of F4–F16	28	103	70	55	59	36	66	51
Friedman rank of F4–F16	1	8	7	4	5	2	6	3
F17	26618.6801	4641277.7674	198099.0415	578588.7550	4299306.6650	86437.0037	143235.1725	81339.7737
F18	2026.3758	121880897.9466	3780.5580	2289.6856	28418.2340	5787.0752	8320.0810	5807.7762
F19	1907.7291	2004.9297	1931.0413	1995.2919	1928.4718	1907.9130	1923.3954	1920.0385
F20	5363.8611	19356.8922	38657.3368	22421.9064	31411.1843	2992.6053	5770.2949	5697.8995
F21	38673.7809	1095231.5294	60455.7923	170612.9594	485593.2936	39074.3102	68597.8240	32529.2852
F22	2481.9864	3134.0717	3072.5807	3161.1458	2722.8879	2346.3986	2496.3689	2483.6063
Friedman value of F17–F22	9	44	32	35	38	14	27	17
Friedman rank of F17–F22	1	8	5	6	7	2	4	3
Overall Friedman value	45	171	114	107	117	65	101	72
Overall Friedman rank	1	8	6	5	7	2	4	3

Table 10 Comparative SD of fitness values of the CEC2014 The results of first six algorithms are as per Zheng [76]

Function	WWO	BA	Hus	GSA	BBO	IWO	SOS	MSOS
F1	244526.8140	104690309.2627	2620084.7953	13187933.1609	16720012.8760	571747.0082	732930.2258	255076.7193
F2	202.2221	7553596375.4800	6012.6897	2903.3044	1549219.3214	8673.4818	20.2822	63.3820
F3	184.6450	17548.6717	540.6109	10432.6453	12764.8742	2692.8884	527.5818	249.8799
F4	36.3636	1973.8532	36.6181	51.5149	38.3545	28.7968	31.7519	40.9633
F5	0.0007	0.0481	0.0783	0.0006	0.0422	0.0038	0.0801	0.0430
F6	2.6204	1.5591	2.1784	1.8319	2.3542	1.1219	2.5681	3.0635
F7	0.0063	32.3193	0.0556	0.0010	0.0264	0.0121	0.0214	0.0271
F8	2.3361	25.6476	12.7304	0.2063	20.6917	10.1117	12.3208	6.9815
F9	11.0977	44.1294	25.9919	17.4329	11.4372	11.3933	24.0796	16.9302
F10	361.6122	518.6548	433.1531	360.9861	0.6800	380.0190	344.1052	187.9776
F11	289.2180	362.2389	465.5429	567.3467	511.5523	447.7160	835.0838	506.0594
F12	0.0561	0.3339	0.0777	0.0010	0.0562	0.0148	0.1833	0.1351
F13	0.0641	0.5483	0.0650	0.0665	0.1061	0.0650	0.0864	0.0783
F14	0.0441	13.9463	0.0474	0.0423	0.1992	0.1191	0.1296	0.0538
F15	0.7753	140338.9490	3.2695	0.7297	4.2976	0.8484	3.7981	4.1851
F16	0.4667	0.1904	0.7249	0.3428	0.5923	0.6144	0.6059	0.7675
F17	12403.5374	1789909.2516	160518.8631	219949.3460	4192494.2708	68473.6644	159023.3392	54940.7907
F18	125.1962	100285357.3457	2246.5148	377.9286	19674.9440	3690.0554	10313.3555	5702.6751
F19	1.3780	20.3164	33.1485	34.3190	27.6885	1.6545	26.6142	22.3884
F20	3177.0847	10283.6255	8492.7252	13860.3564	17604.9005	700.4102	3295.2452	2466.1799
F21	35555.5716	750680.8765	42428.1036	65285.4119	334571.5390	23011.1766	80093.3096	24345.4802
F22	142.8952	205.4095	267.2685	250.0137	234.4393	73.3907	151.5147	141.3538
Overall Friedman value	51.0000	144	113.5	89	128	67.5	108	91
Overall Friedman rank	1	8	6	3	7	2	5	4

problems. Moreover, one can attempt to resolve the persistence problem of tuning of unpredictable parameters in algorithms.

References

1. Baykasoğlu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164. https://doi.org/10.1016/j.asoc.2015.06.056
2. Bingul Z (2007) Adaptive genetic algorithms applied to dynamic multiobjective problems. Appl Soft Comput J 7:791–799. https://doi.org/10.1016/j.asoc.2006.03.001
3. Cheng MY, Prayogo D (2014) Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Comput Struct 139:98–112. https://doi.org/10.1016/j.compstruc.2014.03.007
4. Cheng MY, Prayogo D, Wu YW (2018) Prediction of permanent deformation in asphalt pavements using a novel symbiotic organisms search – least squares support vector regression. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3426-0
5. De Jong KA (1975) An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Comput Commun Sci 56:266
6. Do DTT, Lee J (2017) A modified symbiotic organisms search (mSOS) algorithm for optimization of pin-jointed structures. Appl Soft Comput J 61:683–699. https://doi.org/10.1016/j.asoc.2017.08.002
7. Dorigo M, Maniezzo V, Colorni A (1996) Ant System: Optimization by a Colony of Cooperating Agents. IEEE Trans Syst Man Cybern Part B 26:1–13. https://doi.org/10.1109/3477.484436
8. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. MHS'95 Proc Sixth Int Symp Micro Mach Hum Sci 39–43. https://doi.org/10.1109/MHS.1995.494215
9. Erol OK, Eksin I (2006) A new optimization method: Big Bang-Big Crunch. Adv Eng Softw 37:106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005
10. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
11. Ezugwu AES, Adewumi AO, Frîncu ME (2017) Simulated annealing based symbiotic organisms search optimization algorithm for

traveling salesman problem. Expert Syst Appl 77:189–210. https://doi.org/10.1016/j.eswa.2017.01.053
12. Farshchin M, Camp CV, Maniat M (2016) Multi-class teaching-learning-based optimization for truss design with frequency constraints. Eng Struct 106:355–369. https://doi.org/10.1016/j.engstruct.2015.10.039
13. Fogel DB (1988) An Evolutionary Approach to the Traveling Salesman Problem. Biol Cybern 60:139–144. https://doi.org/10.1007/BF00202901
14. Glover F (1975) Tabu search and adaptive memory programming—advances, applications and challenges. In: In Interfaces in computer science and operations research. pp 1–75
15. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3:95–99. https://doi.org/10.1023/A:1022602019183
16. Gomes HM (2011) Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Syst Appl 38:957–968. https://doi.org/10.1016/j.eswa.2010.07.086
17. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471. https://doi.org/10.1007/s10898-007-9149-x
18. Kaveh A, Farhoudi N (2013) A new optimization method: Dolphin echolocation. Adv Eng Softw 59:53–70. https://doi.org/10.1016/j.advengsoft.2013.03.004
19. Kaveh A, Ilchi Ghazaan M (2015) Hybridized optimization algorithms for design of trusses with multiple natural frequency constraints. Adv Eng Softw 79:137–147. https://doi.org/10.1016/j.advengsoft.2014.10.001
20. Kaveh A, Ilchi Ghazaan M (2017) Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech 228:307–322. https://doi.org/10.1007/s00707-016-1725-z
21. Kaveh A, Jafari L, Farhoudi N (2015) Truss optimization with natural frequency constraints using a dolphin echolocation algorithm. Asian J Civ Eng 16:29–46
22. Kaveh A, Khayatazad M (2013) Ray optimization for size and shape optimization of truss structures. Comput Struct 117:82–94. https://doi.org/10.1016/j.compstruc.2012.12.010
23. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: A novel meta-heuristic method. Comput Struct 139:18–27. https://doi.org/10.1016/j.compstruc.2014.04.005
24. Kaveh A, Mahdavi VR (2015) Two-dimensional colliding bodies algorithm for optimal design of truss structures. Adv Eng Softw 83:70–79. https://doi.org/10.1016/j.advengsoft.2015.01.007
25. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: Charged system search. Acta Mech 213:267–289. https://doi.org/10.1007/s00707-009-0270-4
26. Kaveh A, Zolghadr A (2011) Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm. Asian J Civ Eng 12:487–509
27. Kaveh A, Zolghadr A (2012) Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability. Comput Struct 102–103:14–27. https://doi.org/10.1016/j.compstruc.2012.03.016
28. Kaveh A, Zolghadr A (2013) Topology optimization of trusses considering static and dynamic constraints using the CSS. Appl Soft Comput J 13:2727–2734. https://doi.org/10.1016/j.asoc.2012.11.014
29. Kaveh A, Zolghadr A (2014) Democratic PSO for truss layout and size optimization with frequency constraints. Comput Struct 130:10–21. https://doi.org/10.1016/j.compstruc.2013.09.002
30. Kaveh A, Zolghadr A (2017) Truss shape and size optimization with frequency constraints using Tug of War Optimization. Asian J Civ Eng 18:311–313
31. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by Simulated Annealing. Science 220:671–680. https://doi.org/10.1126/science.220.4598.671
32. Koza JR (1994) Genetic programming as a means for programming computers by natural selection. Stat Comput 4:87–112. https://doi.org/10.1007/BF00175355
33. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798. https://doi.org/10.1016/j.compstruc.2004.01.002
34. Li X, Yin M (2015) Modified cuckoo search algorithm with self adaptive parameter method. Inf Sci (Ny) 298:80–97. https://doi.org/10.1016/j.ins.2014.11.042
35. Liao TW, Kuo RJ (2018) Five discrete symbiotic organisms search algorithms for simultaneous optimization of feature subset and neighborhood size of KNN classification models. Appl Soft Comput J 64:581–595. https://doi.org/10.1016/j.asoc.2017.12.039
36. Miguel LFF, Fadel Miguel LF (2012) Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. Expert Syst Appl 39:9458–9467. https://doi.org/10.1016/j.eswa.2012.02.113
37. Mortazavi A, Toğan V (2017) Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm. Appl Soft Comput J 51:239–252. https://doi.org/10.1016/j.asoc.2016.11.032
38. Noilublao N, Bureerat S (2011) Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms. Comput Struct 89:2531–2538. https://doi.org/10.1016/j.compstruc.2011.08.010
39. Osman IH, Laporte G (1996) Metaheuristics: A bibliography. Ann Oper Res 63:511–623. https://doi.org/10.1007/BF02125421
40. Patel V, Savsani V (2014) Optimization of a plate-fin heat exchanger design through an improved multi-objective teaching-learning based optimization (MO-ITLBO) algorithm. Chem Eng Res Des 92:2371–2382. https://doi.org/10.1016/j.cherd.2014.02.005
41. Piotrowski AP (2013) Adaptive memetic differential evolution with global and local neighborhood-based mutation operators. Inf Sci (Ny) 241:164–194. https://doi.org/10.1016/j.ins.2013.03.060
42. Savsani VJ, Tejani GG, Patel VK (2016) Truss topology optimization with static and dynamic constraints using modified subpopulation teaching–learning-based optimization. Eng Optim 48:1990–2006. https://doi.org/10.1080/0305215X.2016.1150468
43. Savsani VJ, Tejani GG, Patel VK, Savsani P (2017) Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. J Comput Des Eng 4:106–130. https://doi.org/10.1016/j.jcde.2016.10.002
44. Shan H, Yasuda T, Ohkura K (2015) A self adaptive hybrid enhanced artificial bee colony algorithm for continuous optimization problems. BioSystems 132–133:43–53. https://doi.org/10.1016/j.biosystems.2015.05.002
45. Tejani GG, Savsani VJ, Bureerat S, Patel VK (2018) Topology and Size Optimization of Trusses with Static and Dynamic Bounds by Modified Symbiotic Organisms Search. J Comput Civ Eng 32:1–11. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000741
46. Tejani GG, Savsani VJ, Patel VK (2016) Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. J Comput Des Eng 3:226–249. https://doi.org/10.1016/j.jcde.2016.02.003
47. Tejani GG, Savsani VJ, Patel VK (2016) Modified sub-population teaching-learning-based optimization for design of truss structures with natural frequency constraints. Mech Based Des Struct Mach 44:495–513. https://doi.org/10.1080/15397734.2015.1124023
48. Tejani GG, Savsani VJ, Patel VK, Bureerat S (2017) Topology, shape, and size optimization of truss structures using modified teaching-learning based optimization. Adv Comput Des 2:313–331. https://doi.org/10.12989/acd.2017.2.4.313

49. Tollo G di, Lardeux F, Maturana J, Saubion F (2015) An experimental study of adaptive control for evolutionary algorithms. Appl Soft Comput 35:359–372. https://doi.org/10.1016/j.asoc.2015.06.016
50. Tran DH, Cheng MY, Prayogo D (2016) A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time-cost-labor utilization tradeoff problem. Knowledge-Based Syst 94:132–145. https://doi.org/10.1016/j.knosys.2015.11.016
51. Tran DH, Luong-Duc L, Duong MT et al (2018) Opposition multiple objective symbiotic organisms search (OMOSOS) for time, cost, quality and work continuity tradeoff in repetitive projects. J Comput Des Eng 5:160–172. https://doi.org/10.1016/j.jcde.2017.11.008
52. Tejani GG, Pholdee N, Bureerat S, Prayogo D (2018) Multiobjective adaptive symbiotic organisms search for truss optimization problems. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2018.08.005
53. Wang D, Zhang WH, Jiang JS (2004) Truss optimization on shape and sizing with frequency constraints. AIAA J 42:622–630. https://doi.org/10.2514/1.1711
54. Wei L, Tang T, Xie X, Shen W (2011) Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm. Struct Multidiscip Optim 43:665–682
55. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82. https://doi.org/10.1109/4235.585893
56. Yang XS (2009) Firefly Algorithms for Multimodal Optimization. In: In International symposium on stochastic algorithms. pp 169–178
57. Yang XS, Deb S (2009) Cuckoo search via Levy flights. In: 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings. pp 210–214
58. Yu VF, Redi AANP, Yang CL et al (2017) Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem. Appl Soft Comput J 52:657–672. https://doi.org/10.1016/j.asoc.2016.10.006
59. Zuo W, Bai J, Li B (2014) A hybrid OC-GA approach for fast and global truss optimization with frequency constraints. Appl Soft Comput J 14:528–535. https://doi.org/10.1016/j.asoc.2013.09.002
60. Cheng M-Y, Prayogo D, Tran D-H (2016) Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search. J Comput Civ Eng 30:04015036. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000512
61. Duman S (2017) Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones. Neural Comput Appl 28:3571–3585. https://doi.org/10.1007/s00521-016-2265-0
62. Prasad D, Mukherjee V (2016) A novel symbiotic organisms search algorithm for optimal power flow of power system with FACTS devices. Eng Sci Technol Int J 19:79–89. https://doi.org/10.1016/j.jestch.2015.06.005
63. Abdullahi M, Ngadi MA, Abdulhamid SM (2016) Symbiotic Organism Search optimization based task scheduling in cloud computing environment. Futur Gener Comput Syst 56:640–650. https://doi.org/10.1016/j.future.2015.08.006
64. Panda A, Pani S (2016) A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Appl Soft Comput J 46:344–360. https://doi.org/10.1016/j.asoc.2016.04.030
65. Panda A, Pani S (2017) An orthogonal parallel symbiotic organism search algorithm embodied with augmented Lagrange multiplier for solving constrained optimization problems. Soft Comput doi. https://doi.org/10.1007/s00500-017-2693-5
66. Prayogo D (2018) Optimization model for construction project resource leveling using a novel modified symbiotic organisms search. Asian J Civ Eng 3456789:. https://doi.org/10.1007/s42107-018-0048-x
67. Subhabrata Banerjee SC (2017) Power Optimization of Three Dimensional Turbo Code Using a Novel Modified Symbiotic Organism Search. Wirel Pers Commun doi. https://doi.org/10.1007/s11277-016-3586-0
68. Guha D, Kumar P, Banerjee S (2018) Symbiotic organism search algorithm applied to load frequency control of multi-area power system. Energy Syst. https://doi.org/10.1007/s12667-017-0232-1
69. Dosoglu MK, Guvenc U, Duman S, Sonmez Y (2018) Symbiotic organisms search optimization algorithm for economic / emission dispatch problem in power systems. Neural Comput Appl. https://doi.org/10.1007/s00521-016-2481-7
70. Saha S, Mukherjee V (2016) Optimal placement and sizing of DGs in RDS using chaos embedded SOS algorithm. 3671–3680. https://doi.org/10.1049/iet-gtd.2016.0151
71. Zhou Y, Wu H, Luo Q, Abdel-baset M (2018) Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowledge-Based Syst. https://doi.org/10.1016/j.knosys.2018.09.013
72. Prayogo D, Cheng MY, Prayogo H (2017) A Novel implementation of nature-inspired optimization for civil engineering: a comparative study of symbiotic organisms search. 19:36–43. https://doi.org/10.9744/CED.19.1.36-43
73. Jaffel Z, Farah M (2018, March) A symbiotic organisms search algorithm for feature selection in satellite image classification. In: Advanced Technologies for Signal and Image Processing (ATSIP), 2018 4th International Conference on(pp. 1–5). IEEE
74. Sulaiman M, Ahmad A, Khan A, Muhammad S (2018) Hybridized symbiotic organism search algorithm for the optimal operation of directional overcurrent relays. Hindawi Complex 2018:11. https://doi.org/10.1155/2018/4605769
75. Tejani GG, Savsani VJ, Patel VK, Mirjalili S (2018) Knowledge-Based Systems Truss optimization with natural frequency bounds using improved symbiotic organisms search. Knowledge-Based Syst 5:1–17. https://doi.org/10.1016/j.knosys.2017.12.012
76. Zheng Y (2015) Computers & Operations Research Water wave optimization: A new nature-inspired metaheuristic. Comput Oper Res 55:1–11. https://doi.org/10.1016/j.cor.2014.10.008
77. Črepinšek M, Liu S-H, Mernik M (2013) Exploration and Exploitation in Evolutionary Algorithms: A Survey. ACM Comput 45:1–33. https://doi.org/10.1145/2480741.2480752
78. Geem Z, Kim J, Loganathan GV (2001) A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76:60–68. https://doi.org/10.1177/003754970107600201
79. Al-sharhan S (2016) An enhanced symbiosis organisms search algorithm: an empirical study. Neural Comput Appl doi. https://doi.org/10.1007/s00521-016-2624-x
80. Liang JJ, Qu BY, Suganthan PN (2014) Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization
81. Grandhi R (1993) Structural Optimization with Frequency Constraints — A Review. AIAA J 31:2296–2303. https://doi.org/10.2514/3.11928
82. Kaveh A, Ilchi M, Taha G (2013) An improved ray optimization algorithm for design of truss structures. Period Polytech 2:97–112. https://doi.org/10.3311/PPci.7166
83. Kaveh A, Ghazaan MI (2014) Enhanced colliding bodies algorithm for truss optimization with frequency constraints. J Comput Civ Eng 29:1–11. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000445
84. Miao F, Zhou Y, Luo Q (2018) A modified symbiotic organisms search algorithm for unmanned combat aerial vehicle route planning problem. J Oper Res Soc 5682:1–32. https://doi.org/10.1080/01605682.2017.1418151

85. Bureerat S, Ph D, Pholdee N, Ph D (2015) Optimal Truss Sizing Using an Adaptive Differential Evolution Algorithm. J Comput Civ Eng. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000487
86. Ou-yang C, Hanyata TB, Samadhi TMAA (2015) Hybrid self-adaptive-velocity particle swarm optimisation-Cooper heuristic for the facility location allocation problem in Jakarta. https://doi.org/10.1080/23302674.2015.1029565
87. Pham HA (2016) Advances in Engineering Software Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison. Adv Eng Softw 102:142–154. https://doi.org/10.1016/j.advengsoft.2016.10.004
88. Lieu QX, Do DTT, Lee J (2018) An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 195:99–112. https://doi.org/10.1016/j.compstruc.2017.06.016
89. Ho-Huu V, Nguyen-Thoi T, Truong-Khac T, Le-Anh L, Vo-Duy T (2016) An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. Neural Comput Appl 29:167–185. https://doi.org/10.1007/s00521-016-2426-1
90. Kaveh A, Javadi SM (2013) Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm. 1605:1595–1605. https://doi.org/10.1007/s00707-013-1006-z
91. Kaveh A, Zolghadr A (2017) Cyclical parthenogenesis algorithm for layout optimization of truss structures with frequency constraints. Eng Optim 0:1–18. https://doi.org/10.1080/0305215X.2016.1245730
92. Kaveh A, Zolghadr A (2014) A new PSRO algorithm for frequency constraint truss shape and size optimization. Struct Eng Mech 52:445–468. https://doi.org/10.12989/sem.2014.52.3.445
93. Jalili S, Talatahari S (2017) Optimum Design of Truss Structures Under Frequency Constraints using Hybrid CSS-MBLS Algorithm. 00:1–14. https://doi.org/10.1007/s12205-017-1407-y