ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Design and Engineering ▮ (▮▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/jcde

Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints

Vimal J. Savsani[a,*], Ghanshyam G. Tejani[a,b], Vivek K. Patel[a], Poonam Savsani[a]

[a]Department of Mechanical Engineering, Pandit Deendayal Petroleum University, Raysan, Gandhinagar, Gujarat, India
[b]Department of Mechanical Engineering, School of Engineering, RK University Rajkot, Gujarat, India

Received 22 June 2016; received in revised form 24 October 2016; accepted 28 October 2016

Abstract

In this paper, simultaneous size and topology optimization of planar and space trusses subjected to static and dynamic constraints are investigated. All the benchmark trusses consider discrete cross-sectional areas to consider the practical aspect of manufacturing. Moreover, Trusses are considered with multiple loading conditions and subjected to constraints for natural frequencies, element stresses, nodal displacements, Euler buckling criteria, and kinematic stability conditions. Truss topology optimization (TTO) can be accomplished by the removal of superfluous elements and nodes from the highly hyper static truss also known as the ground structure and results in the saving of the mass of the truss. In this method, the difficulties arise due to the singular solution and unnecessary analysis; therefore, FEA model is reformed to resolve these difficulties.

The static and dynamic responses to the TTO problems are challenging due to its search space, which is implicit, non-convex, non-linear, and often leading to divergence. Modified meta-heuristics are effective optimization methods to handle such problems in actual fact. In this paper, modified versions of Teaching–Learning-Based Optimization (TLBO), Heat Transfer Search (HTS), Water Wave Optimization (WWO), and Passing Vehicle Search (PVS) are proposed by integrating the random mutation-based search technique with them. This paper compares the performance of four modified and four basic meta-heuristics to solve discrete TTO problems.
© 2016 Society of CAD/CAM Engineers. Publishing Servies by Elsevier. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Teaching–Learning-Based Optimization; Water Wave Optimization; Passing Vehicle Search; Structural optimization; Discrete optimization; Heat Transfer Search

1. Introduction

The majority of the Truss Topology Optimization (TTO) problems reported in the literature have been considered with only stress and displacement constraints. Yet, few studies have been covered by considering frequencies and buckling constraints along with stress and displacement constraints [1-5]. The natural frequencies of an engineering structure are an essential parameter when such structure is subjected to the dynamic excitations [1,6,7]. Many engineering structures are subjected to dynamic excitation due to the working condition and certain unpredicted circumstances that may lead to unwanted vibrations [8]. Such a state becomes dangerous if the dynamic responses produce resonance; therefore, some convinced restrictions should be enforced on natural frequencies to protect an engineering structure [2,9]. Moreover, frequency constraints increase the complexity of the TTO problems [10]. Buckling can also have consequence effect and it includes additional complexity, which makes the TTO problems more challenging [11-17]. Moreover, simultaneous consideration of natural frequencies and buckling constraints adds more limitations to the TTO problems [1]. On the contrary, these constraints cannot be ignored in order to assure practicability of a structure. Kinematic instable and invalid

*Corresponding author.
E-mail addresses: vimal.savsani@gmail.com (V.J. Savsani), p.shyam23@gmail.com (G.G. Tejani), viveksaparia@gmail.com (V.K. Patel), poonam.savsani@gmail.com (P. Savsani).
Peer review under responsibility of Society of CAD/CAM Engineers.

http://dx.doi.org/10.1016/j.jcde.2016.10.002
2288-4300/© 2016 Society of CAD/CAM Engineers. Publishing Servies by Elsevier. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Please cite this article as: Savsani VJ, et al. Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Designand Engineering* (2016), http://dx.doi.org/10.1016/j.jcde.2016.10.002

structures are key obstacles in the path of the TTO, therefore such structures need to identify and handle efficiently to avoid a large number of unwanted analyses.

Referable to the presence of multiple loading along with stress, displacement, buckling, frequency, kinematic stability constraints, and mixed variables the TTO problems become more challenging for optimization methods. Therefore, the search becomes implicit, nonlinear, and non-convex. Also, the TTO problem may not provide gradient-based information, hence gradient based methods may not converge to the global solution. Thus, effective optimization methods are needed to resolve such problems and researchers are constantly investigating in this arena. On the other hand, meta-heuristics do not require gradient-based information, thus non-differentiable problems can be solved. In addition, meta-heuristics can work effectively with large, highly nonlinear, and non-convex search space can handle problems with multiple variables, and can achieve the global solution. However, the meta-heuristics are subjected to low convergence and consume high computational time. Furthermore, investigation of various meta-heuristics and its improvement is a growing area of interest and many satisfactory optimization results have been reported in the literature ([18-23,6,7]). The performance of a meta-heuristic can be enhanced either by modifying its features or through exert the merits of different original methods. Modification of a meta-heuristic set a good balance between exploration and exploitation so as to maintain diversity in the population, improve robustness, and convergence rate of the algorithm. Modified meta-heuristics have been proven for its outstanding reliability and efficiency to solve engineering problems, and became a striking alternative of basic algorithms. This characteristic is a strong motivation for the implementation of modified meta-heuristics in this study.

The progression of the Teaching–Learning-Based Optimization (TLBO) [24,25], Heat Transfer Search (HTS) [26], Water Wave Optimization (WWO) [27], and Passing Vehicle Search (PVS) [28] algorithms depend on characteristics of the initial as well as current population/solutions. The population may change to a small extent when the mean of the solutions is near to the best solution of the population and same is true if the current solution is close to the other randomly selected solution during the course of optimization. This state results in a premature convergence to a local optimal solution as solutions remains almost close to each other. The state in which the population does not improve further may be considered as a trapped local optimum. In such instance, the solution should be changed to search a better solution and to avoid local optima traps. Therefore, a random mutation-based search technique is introduced to answer the stated issue. Random changes to design variables of the solution are called mutation and it has been used effectively in many studies [29-50]. In a proposed random mutation-based search, mutation vector is generated by a fusion of host design variables and randomly generated variables; therefore it improves the exploration of the search space. The highly heuristic nature of the random mutation-based search phase permits search to jump into non-visited regions (exploration). In this way, this phase has an additional characteristic to avoid local optima trap of the parent meta-heuristic. Moreover, the HTS, WWO, and PVS algorithms are recently developed algorithms and it is required to explore such methods for the challenging problems of structural optimization, whereas the TLBO algorithm is an effective technique and has an effective impact for different engineering optimization problems. Moreover, the considered meta-heuristics have distinct search mechanisms and it is virtually impossible to forecast the influence of the modification for each of the applications and meta-heuristics. Thus, the TLBO, HTS, WWO, and PVS algorithms are modified with the random mutation-based search technique to propose a Modified TLBO (MTLBO), Modified HTS (MHTS), Modified WWO (MWWO), and Modified PVS (MPVS) in this paper.

In this paper, three distinct trusses are investigated by using four basic meta-heuristics (viz., TLBO, HTS, WWO, and PVS) and four modified meta-heuristics (viz., MTLBO, MHTS, MWWO, and MPVS) by considering the overall weight as an objective function and stress, displacement, buckling, frequency, and kinematic stability as constraints. In addition, discrete cross-sectional areas are considered as design variables to ensure practicability of the structure.

2. Structural optimization

Structural optimization can be classified into three groups: size optimization, shape optimization, and topology optimization [51]. Size optimization deals to find the elemental cross-sectional areas that contribute to the minimization of the optimization function like weight, cost, etc. Topology optimization requires more computational efforts because it deals with all the generated different topologies rather than a particular topology and results in a great saving of weight by searching finest topology [31,22,23]. Many researchers continuously put their efforts to investigate various optimization methods to tackle structural optimization problems.

A ground structure is set of all probable connections among nodes and these networks can be controlled by restricting element length. In the ground structure method, the operation for element removal and addition is performed such that it leads to topology optimization [52,53]. Due to removal or addition of elements, the topology may result in a singular topology solution, which is a challenging issue of the TTO. One way to avoid singularity is to assign tiny values of the removed elements [54] but it adds unnecessary analysis due to the removed element being there in a form of microelement (element with the negligible area). To overcome the above limitations this study considers restructuring of Finite Element Analysis (FEA) model. In this method, the FEA model is restructured by removing the connectivity to the node if deleted. This method avoids unnecessary analysis of removed elements and nodes; also, it avoids the singularity. Most of the study neglects the mass at the node; however, it is necessary to consider node weight because it has an effect on the overall weight of a structure [29].

Many studies are reported for the size optimization with various constraints such as stress, displacement, buckling, frequency, and kinematic stability. Likewise, size, shape, and/or topology optimization with only stress and displacement constraints is reported by many researchers [33,3,31,29,55,48]. Size and shape optimization with only frequency constraints is investigated by several researchers ([3,56,57,6,7]). Many studies are reported for the size and shape optimization with frequency constraints, but on the other hand, structural optimization with simultaneous static and dynamic constraints has been covered by a few researchers [1,58,2,3,4,54,59,5].

Two different approaches are practiced for the TTO, depending on the different steps followed for the optimization problems such as two-stage approach and single stage (simultaneous) approach. In two-stage approach, a set of feasible topologies is investigated by considering constant cross-sectional areas during the first stage and topologies recognized during the first stage are optimized for size in the second stage [1,2,60]. Two-stage optimization approach may fail to reach the global optimum solution if feasible topologies identified in the first stage are not having an optimal topology. Single stage optimization approach requires more computational efforts because it deals with simultaneous size and topology optimization [31]. In this paper, Single stage optimization approach with ground structure method and restructuring of FEA model are used for the size and topology optimization.

3. Optimization algorithms

This comparative study comprises four basic optimization algorithms, and four modified algorithms to investigate the TTO problems. All the considered optimization algorithms are population-based meta-heuristics. These algorithms set off with a randomly generated population also known as set of solutions. The population is then updated by using a succession of different mathematical formulas, which are primarily inspired by some natural law. In addition, the proposed algorithms use greedy selection to select whether to retain the current or modified solution. It should be noted that this study does not include the removal of duplicate solutions in the TLBO, HTS, WWO, and PVS algorithms, as it requires additional function evaluations (FE). The structure of all modified algorithms are briefly summarized in the subsequent sections:

3.1. The random mutation-based search

Mutation operators comprise a random alteration of design variables during the course of optimization. The role of mutation on various meta-heuristics has been reviewed comprehensively by [32]. They also explored the effectiveness and importance of new mutation operators. [29] used a binary mutation operator in GA, whereas [30] proposed genetic adaptive search by using polynomial mutation operator. [33] used m-point-mutation of the binary structure vector to build-up topological optimal structure, whereas [31] studied a parameter-based mutation operator in real-coded GA for the TTO. Multi-objective optimization of space truss using a random design variable type mutation on GA is investigated by [34]. [35] experimentally demonstrated that mutation-oriented multi-objective evolutionary algorithms are a better performer than their crossover-oriented versions for topology optimization of a continuous structure. [36] used global and local neighborhood-based mutation operators in Differential Evolutions (DE), however, original DE is based on a single point mutation strategy. [38] offered the particle-position-resetting approach, inspired by a mutation in GA, to accelerate convergence and to avoid a local optimum. 'Raining process' phase of water cycle algorithm acts similar to mutation operator in GA [37]. Moreover, the parasitism phase of the symbiotic organisms search [9] is an effective usage of random mutation search. Biogeography-based optimization algorithm also works on mutation operator phase [40,61]. [41] used a modified mutation operation in order to eliminate the duplicate solutions in the TLBO algorithms. [42] used a Gaussian mutation in the gravitational search algorithm. [43] investigated the effectiveness of DE-mutation-strategies on tri-population approach as hybrid DE and practical swarm optimization. [44] proposed a directional mutation operator for differential evolution. [45] applied a hybrid mutation operator on DE. [46] balanced ensemble of three mutation strategies for multi-population DE with on large-scale global optimization problems. [47] used mutation scaling factor, which automatically adjusts the values of controlling parameters to propose a sinusoidal DE. [48] proposed two new mutation rules based on the 'rand' and 'best' individuals of the population to propose a modified cuckoo search algorithm. [49] introduced a species mutation technique, which is a combination of a neighbor mutation and uniform mutation in order to improve the accuracy of solutions for TTO. This strategy has the advantage of global and local search. Moreover, four mutation operators are tested by [50] in order to enhance the performance of adaptive DE algorithm with optional external archive algorithm.

A mutation operator can help the algorithm to lead the population toward the global optimum instead of becoming trapped in local optima. Mutation is a powerful strategy to increase search diversity. In this aspect, a random mutation-based search is proposed with TLBO, HTS, WWO, and PVS algorithms in order to enhance its effectiveness and to investigate its consequence on structural optimization problems.

In random mutation-based search, a mutation vector (X_i') of the *i*th population member is produced by mutating it's *j*th selected design variables ($j=1,2,\ldots,m$). Thus, the randomly selected design variables are modified using a randomly generated number within its bounds. In this way, mutation vector is a fusion of design variables of the *i*th population member and a randomly generated design variable. In the next step, the functional value of a randomly selected *k*th

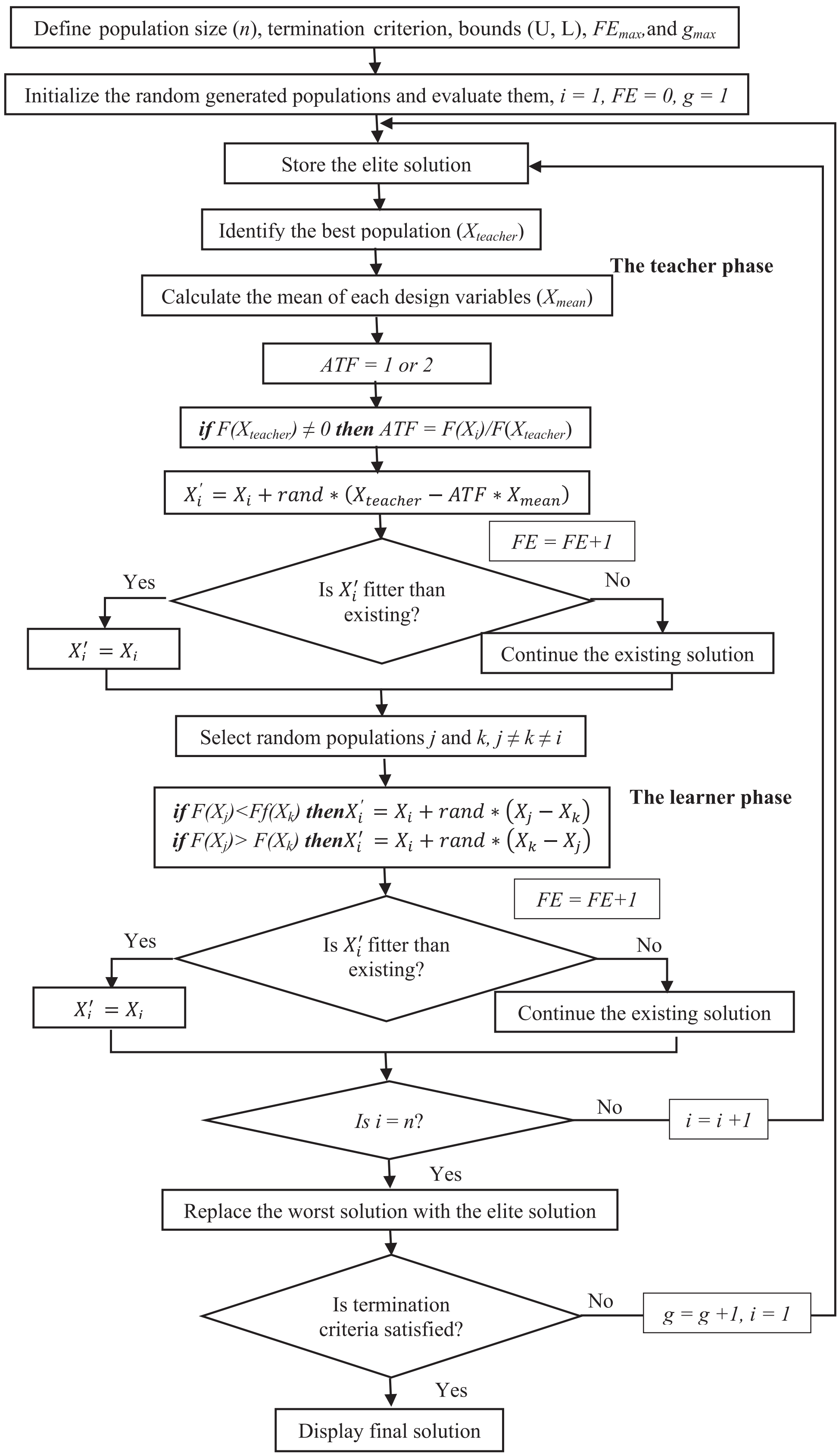

Fig. 1. Flow chart of the TLBO algorithm.

Please cite this article as: Savsani VJ, et al. Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Designand Engineering* (2016), http://dx.doi.org/10.1016/j.jcde.2016.10.002

population member ($k \neq i$) is compared with the functional value of the mutation vector. If mutation vector has better functional value than the kth population member, it will eliminate kth population member and acquire its position. Therefore, the random mutation-based search improves the exploration and exploitation of the search space as mutation vector is generated by a fusion of the host design variable and randomly generated variable. The highly random nature of the phase allows the search to jump into non-visited regions (exploration) so as to escape premature convergence and also permits a local search of visited regions (exploitation) to improve convergence rate. In this way, this phase improves exploration and exploitation capabilities of the parent meta-heuristic. Thus, the TLBO, HTS, WWO, and PVS algorithms are modified with the mutation vector in order to improve their performance. The detail pseudo code to generate the mutation vector of ith population is given as follows:

```
Select X_i
  for j=1:m do
    Generate a random number (r_j)
      if r_j=0 then
        X'_{i,j} = L_{i,j} + R * (U_{i,j} - L_{i,j})
    end if
  end for
where X_i is the ith population member, j
indicate the position of design variable, r_j
is a random integer number over [0, 1], R is a
random number over [0, 1], X' is a mutation
vector, L and U are lower and upper limit on
design variables respectively.
```

3.2. TLBO

The TLBO algorithm is a meta-heuristic, proposed by [24,25], which is based on the influence of a teacher on the outcomes of learners in a classroom. The classroom teaching–learning is one sort of essential track where students learn from the teacher and follow learners to improve their knowledge. The TLBO algorithm is a population-based algorithm, where the learners are viewed as the population and various subjects offered to the learners are considered as the design variables.

The TLBO algorithm initiates with a randomly generated population, where the class has 'n' number of students (i.e., population size) studying 'm' number of the subjects (i.e., design variables). In the following stage, the population is updated by the teacher phase and learner phase in each generation. Moreover, the updated solution in the TLBO algorithm is accepted only if it has a better function value. Subsequently, a worst solution of the population is replaced by the elite solution. The TLBO algorithm is explained with the aid of flow chart shown in Fig. 1.

3.3. MTLBO

The TLBO algorithm works on the teacher phase and the learner phase. It can be understood from the teacher phase that the updated population has a great influence of the best population member considered as a teacher ($X_{teacher}$) and mean of a population (X_{mean}). The population is updated by interactions among the randomly selected population during the learner phase. In the progression of the optimization process, the population might change very small or retain its current position, when the population is close to the best solution ($X_{teacher}$) or mean of a population (X_{mean}) and even if a population (X_i) is close to the other randomly selected population member (X_j, X_k). This condition may end in premature convergence as the population remains almost close to the earlier value. Thus, such condition results in premature convergence and needs to address effectively.

It is observed from the literature that the TLBO algorithm can be made more effective either by modification or hybridization [18,20,21,61,7]. This feature encouraged us to formulate the MTLBO algorithm and to investigate its effect on the TTO problems. The main contribution of this study is to propose modification strategy that is based on the mutation vector in order to diminish premature convergence. The flow chart of the MTLBO algorithm with the stepwise procedure is shown in Fig. 2. It can be seen from the flow chart that the population is updated in each generation by the teacher phase, the learner phase of the TLBO algorithm as the main search procedure followed by the random mutation-based phase.

3.4. HTS

The HTS algorithm, proposed by [26], works on heat transfer due to interaction within the system molecules as well as with the surrounding in order to reach a thermal equilibrium. The natural law of thermodynamics states that "Any system always try to achieve equilibrium state with its surroundings". Therefore, thermodynamically imbalance system always tries to achieve thermal equilibrium by heat transfer between the system and its surrounding. The modes of heat transfer are conduction, convection, and radiation that plays an important role in setting a thermal equilibrium. Thus, the HTS algorithm considers 'the conduction phase', 'the convection phase', and 'the radiation phase' to reach an equilibrium state. In the HTS algorithm, all three modes of heat transfer have an equal chance to transfer heat and one of the heat transfer mode is decided randomly for each generation.

The HTS algorithm initiates with a randomly generated population, where the system has 'n' number of molecules (i.e., population size) and the temperature level (i.e., design variables) is 'm'. In the next stage, the population is updated in each generation by one of the randomly selected heat transfer mode. Moreover, the updated solution in the HTS algorithm is accepted only if it has a better functional value. Subsequently, worst solutions of the population are replaced by the elite

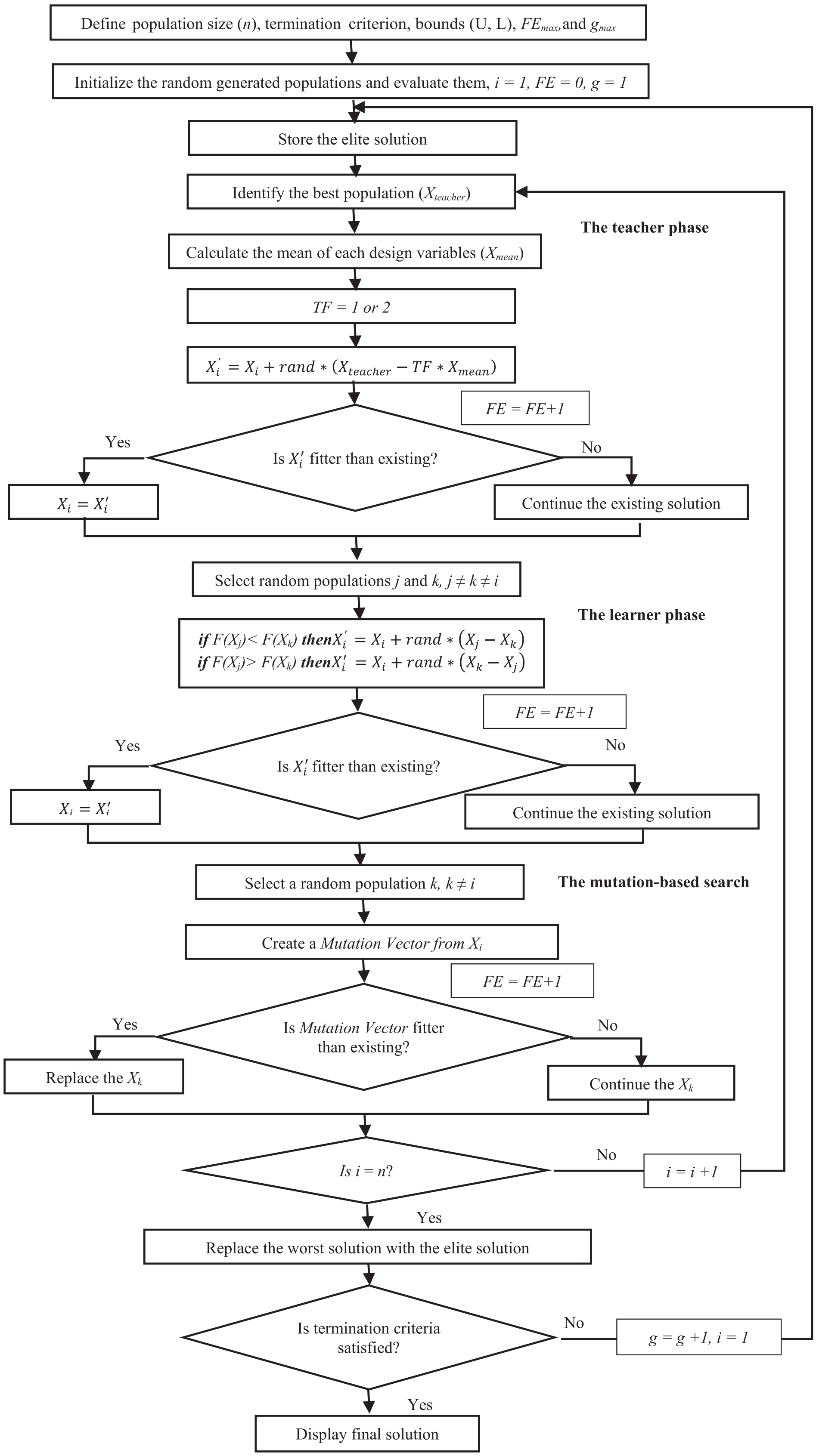

Fig. 2. Flow chart of the MTLBO algorithm.

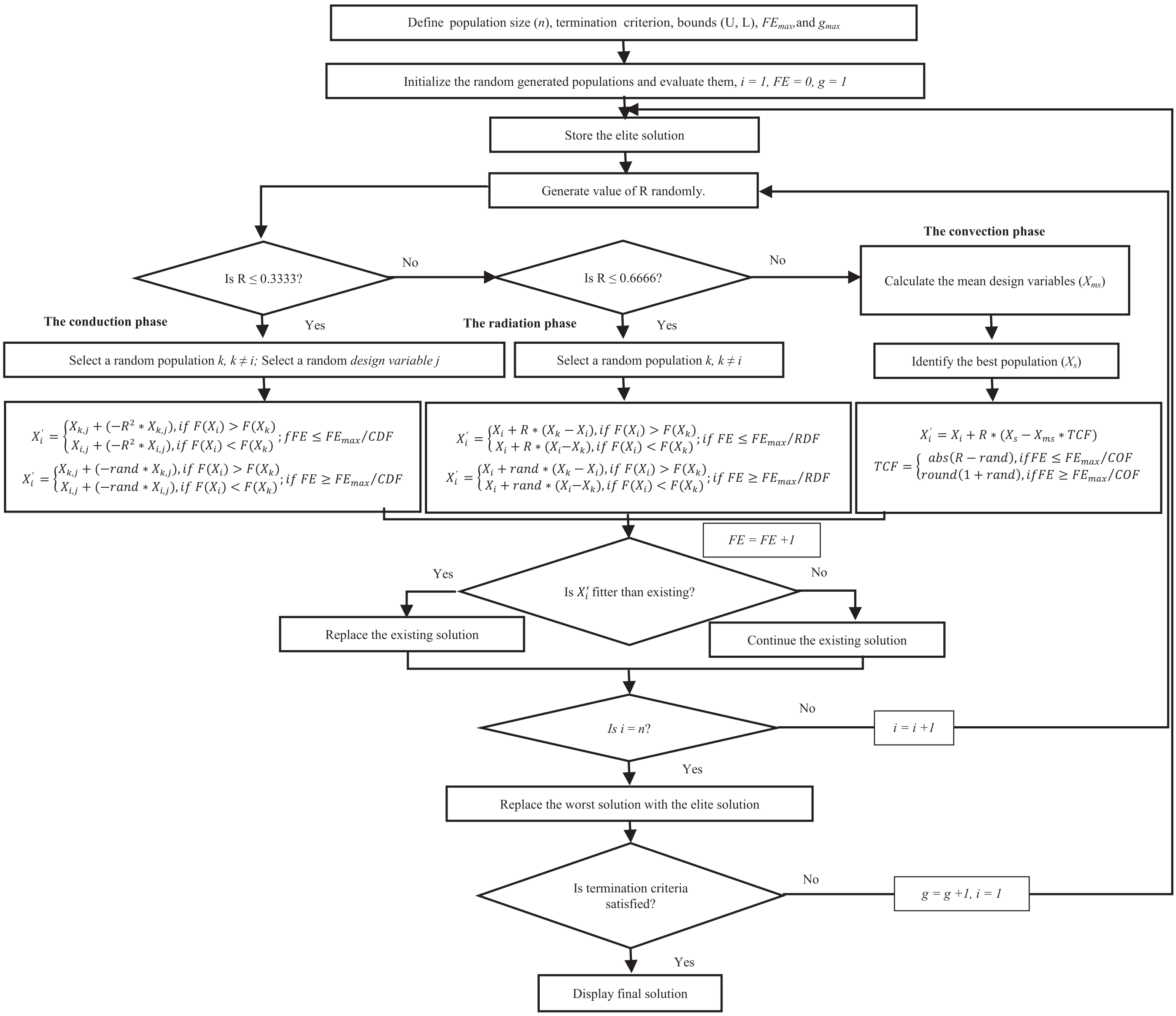

Fig. 3. Flow chart of the HTS algorithm.

solutions. The conduction factor (CDF), the convection factor (COF), and the radiation factor (RDF) are assumed to be 2, 10, and 2 respectively as per [26]. The HTS algorithm is explained with the aid of a flow chart shown in Fig. 3.

3.5. MHTS

As discussed earlier the HTS algorithm works on 'the conduction phase', 'the convection phase', and 'the radiation phase'. The mathematical formulation of each of the phases is given in Fig. 3. It can be understood from each of the phases that the population may change to a small extent, when the population is close to the mean of a population (the mean temperature, X_{ms}) or the best population member (the surrounding temperature, X_s) and even if a population (X_i) is close to the other randomly selected population (X_k) during the course of optimization. This condition may result in premature convergence as the population remains almost close to the earlier value.

Many studies have been reported for the modified versions of various meta-heuristics to improve the effectiveness; however, the modification of HTS with different strategies is still under research. Therefore, the efforts must be put in to modify HTS which can make it effective, robust, and able to provide accurate solutions. In this aspect, the HTS algorithm is modified to a new modified meta-heuristic called the MHTS algorithm. The main contribution of this study is to propose modification strategy that is based on the mutation vector in order to diminish premature convergence. The flow chart of the MHTS algorithm with the stepwise procedure is shown in Fig. 4. It can be seen from the flow chart that the population is updated in each generation by one of the randomly selected heat transfer mode and the random mutation-based phase.

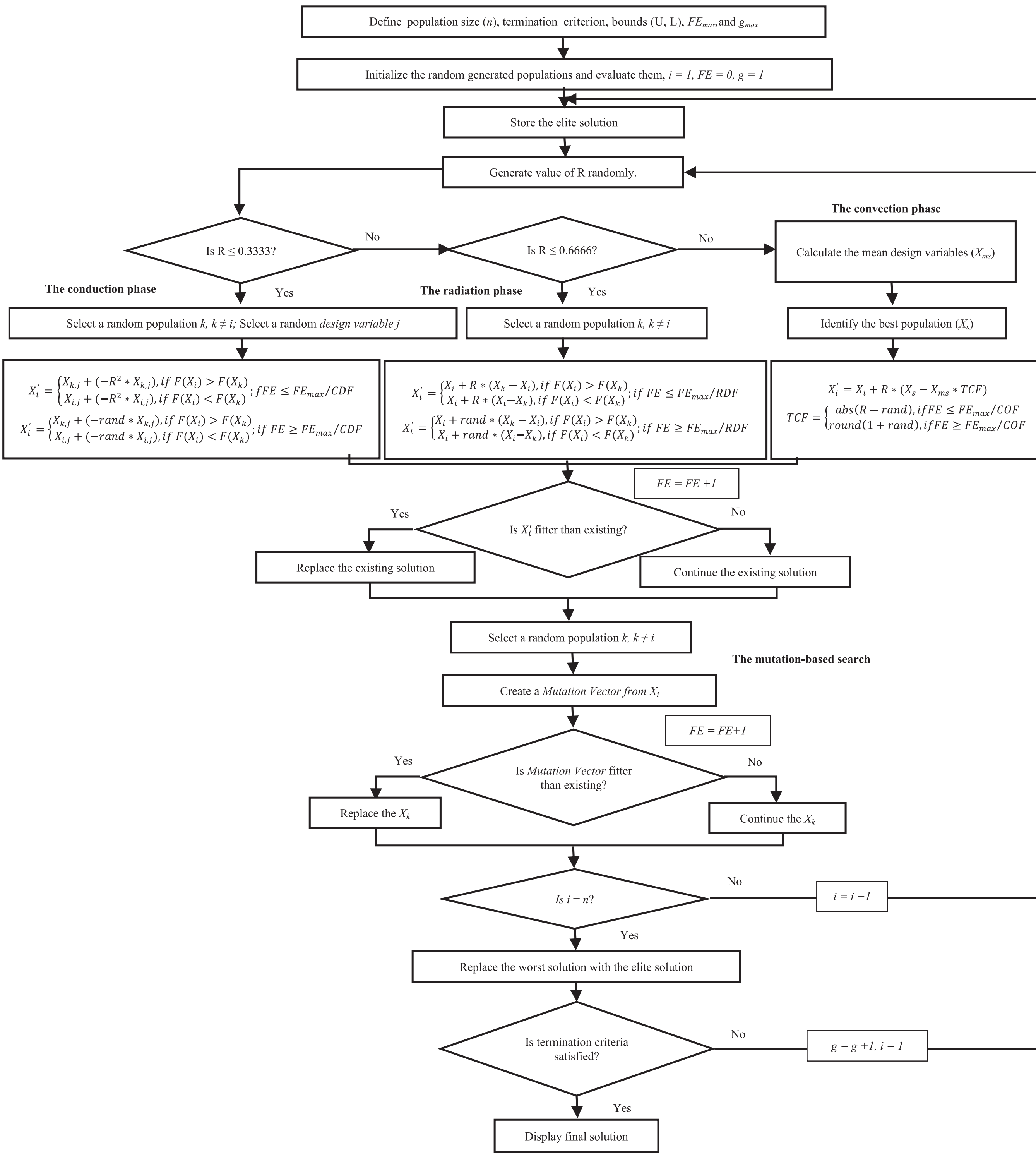

Fig. 4. Flow chart of the MHTS algorithm.

3.6. WWO

The WWO algorithm, proposed by [27], is a meta-heuristic inspired from the shallow water wave theory. The WWO algorithm mimics phenomena of water wave motion, such as propagation, refraction, and breakage. In this algorithm, low energy water waves have large wavelengths that allow making exploration capability stronger during short life cycles, whereas high energy waves have small wavelengths that allow making exploration capability stronger during long life cycles. Moreover, the progression of the WWO algorithm is controlled by various parameters viz., the wavelength (λ), the maximum wave height (h_{max}), the wavelength reduction coefficient (α), the breaking coefficients (β_{max} and β_{min}), and the maximum number of breaking directions (k_{max}).

The WWO algorithm initiates with a randomly generated population (i.e., waves). In the next stage, the population is updated in each generation by 'propagation operator', 'refraction operator', and 'breaking operator' respectively. Subsequently, worst solutions of the population are replaced by the elite solutions. The process is repeated until it satisfies the termination criterion. The WWO algorithm is explained with the aid of a flow chart in Fig. 5.

3.7. MWWO

The WWO algorithm works on 'propagation operator', 'refraction operator', and 'breaking operator'. It can be seen from mathematical formulation, shown in Fig. 5, that the population and its functional values ($F(X)$, $F(X')$) might change very small, when the population (X_i) is close to the mean a population (X_{mean}) or the best population member (X_{best}) and even if the population (X_i) is close to the worse population member (X_{worse}) during the course of optimization. Moreover, the WWO algorithm controlling operators also depend on the population and their functional values (i.e., X_{mean}, X_{best}, X_{worse}, $F(X)$, $F(X')$, etc.). This phenomena may result in premature convergence to a local optimal.

[19] have proposed modified version of the WWO algorithm using variable population size to provide a better tradeoff between exploration and exploitation, however, the modification of the WWO algorithm with other strategies is still under research. Therefore, the efforts must be put in to modify WWO algorithm with other optimization techniques in order to get more effective, robust, and accurate solutions. In this aspect, the WWO algorithm is modified to a new modified meta-heuristic called MWWO. The MWWO algorithm incorporates mutation vector as discussed earlier. The flow chart of the MWWO algorithm with the stepwise procedure is shown in Fig. 6. It can be seen from the flow chart that the population is updated in each generation by one of the randomly selected water wave motion operator followed by the random mutation-based search.

3.8. PVS

The PVS algorithm, proposed by [28], is a novel meta-heuristic algorithm. The PVS algorithm mimics the vehicle passing mechanism on a two-lane highway. The most important criteria is to have a safe overtaking opportunity (passing) in a two-lane vehicle passing mechanism. This mechanism depends on many complex, interdependent parameters such as availability of gaps in the opposing traffic stream, speed, and acceleration of individual vehicles, traffic, and a driver's skill, as well as road and weather conditions. The PVS algorithm considers three types of vehicles (viz., Back Vehicle (BV), Front Vehicle (FV), and Oncoming Vehicle (OV)) on a two-lane highways, which are responsible for the passing mechanism. BV intends to pass FV, however, it is only possible if FV speed is slower as compared to BV. If FV speed is higher as compared to BV, then no passing is possible. Moreover, passing depends on the position and speed of OV, and also on the distance between them and their velocities. Therefore, [28] considered various conditions as follows:

Assume three different vehicles (BV, FV, and OV) on a two-lane highway having different velocities (V_1, V_2, and V_3) with x is the distance between BV and FV, and y is the distance between FV and OV at any particular time instance. This results in two primary conditions based on the velocity of FV and BV, i.e., FV is slower than BV ($V_1 > V_3$) and vice versa. If FV is faster than BV, then no passing is possible and BV can move with its desired velocity. Passing is possible only if FV is slower than BV. In this situation also, overtaking is only possible, if the distance from the FV at which overtaking occurs is less than the distance travelled by OV. Therefore, different conditions arise for the selected vehicles. The PVS algorithm with its different situations is explained with the aid of a flow chart in Fig. 7.

3.9. MPVS

The modifications of the PVS algorithm with different strategies is not reported in the literature yet as it is a very recent algorithm and research is still going on to develop its modified variants. Therefore, the efforts must be put in to modify the PVS algorithm with other optimization techniques, which can be more effective, robust, and able to provide accurate solutions. It can be seen from the mathematical formulation, illustrated in Fig. 7, that the population may change to a small extent or retain its current position when the distance between a population (X_i) to the other randomly selected population (X_k or X_l) is small. This state may result in a local optima stuck as the new population (X') may not improve further. In this aspect, The random mutation-based search is incorporated into the PVS algorithm to develop the MPVS algorithm. The MPVS algorithm incorporates mutation vector as discussed earlier. The flow chart of the MPVS algorithm with the stepwise procedure is illustrated in Fig. 8. It can be seen from the flow chart that the population is updated by various conditions of the PVS algorithm followed by the random mutation-based search.

As all the algorithms are modified by adding random mutation strategies, there is a change in the computational complexity of the algorithms. The complexity of an optimization algorithm can be characterized as time complexity and the computational complexity. This can be addressed as follows:

a. Time complexity: this type of complexity deals with the time required by each mechanism such as generation of the initial population, updating solution, selection of updated solution, etc., and total time (which is the sum of all such measures). Any modification in the basic algorithm may change such measures.

   In this study, time consumption for the generation of the initial population is identical for all the algorithms as it works on an identical method to initialize the randomly

ARTICLE IN PRESS

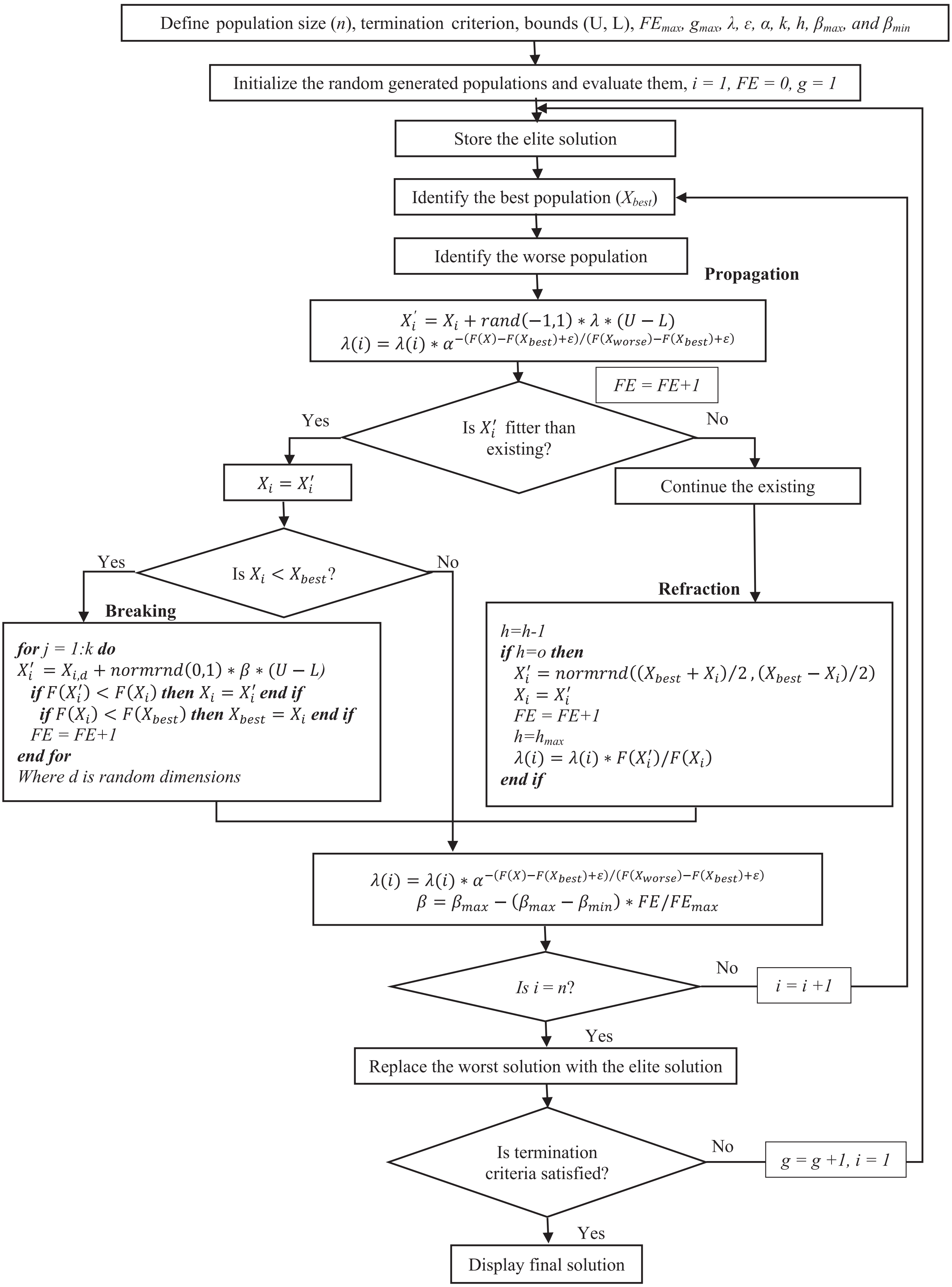

Fig. 5. Flow chart of the WWO algorithm.

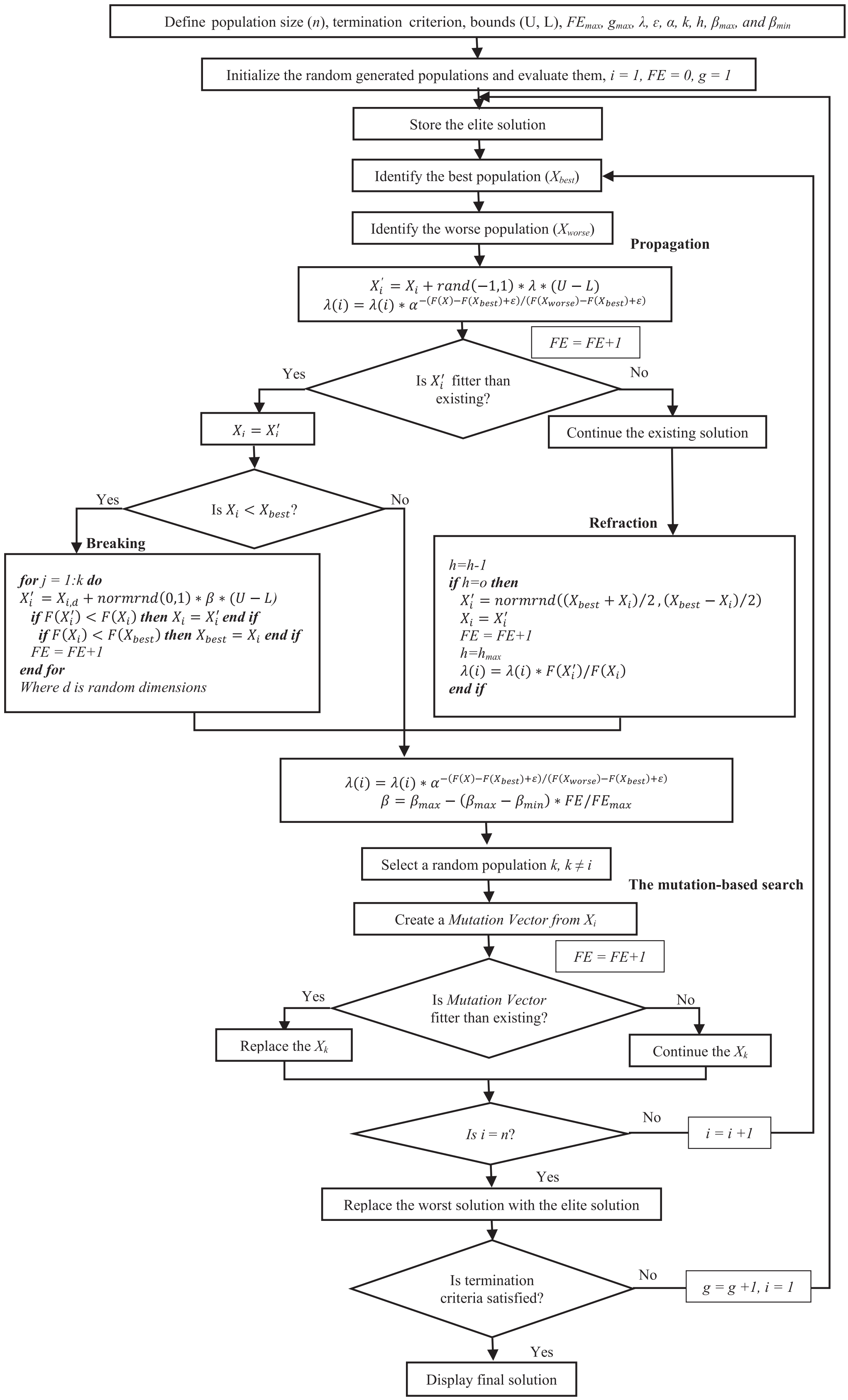

Fig. 6. Flow chart of the MWWO algorithm.

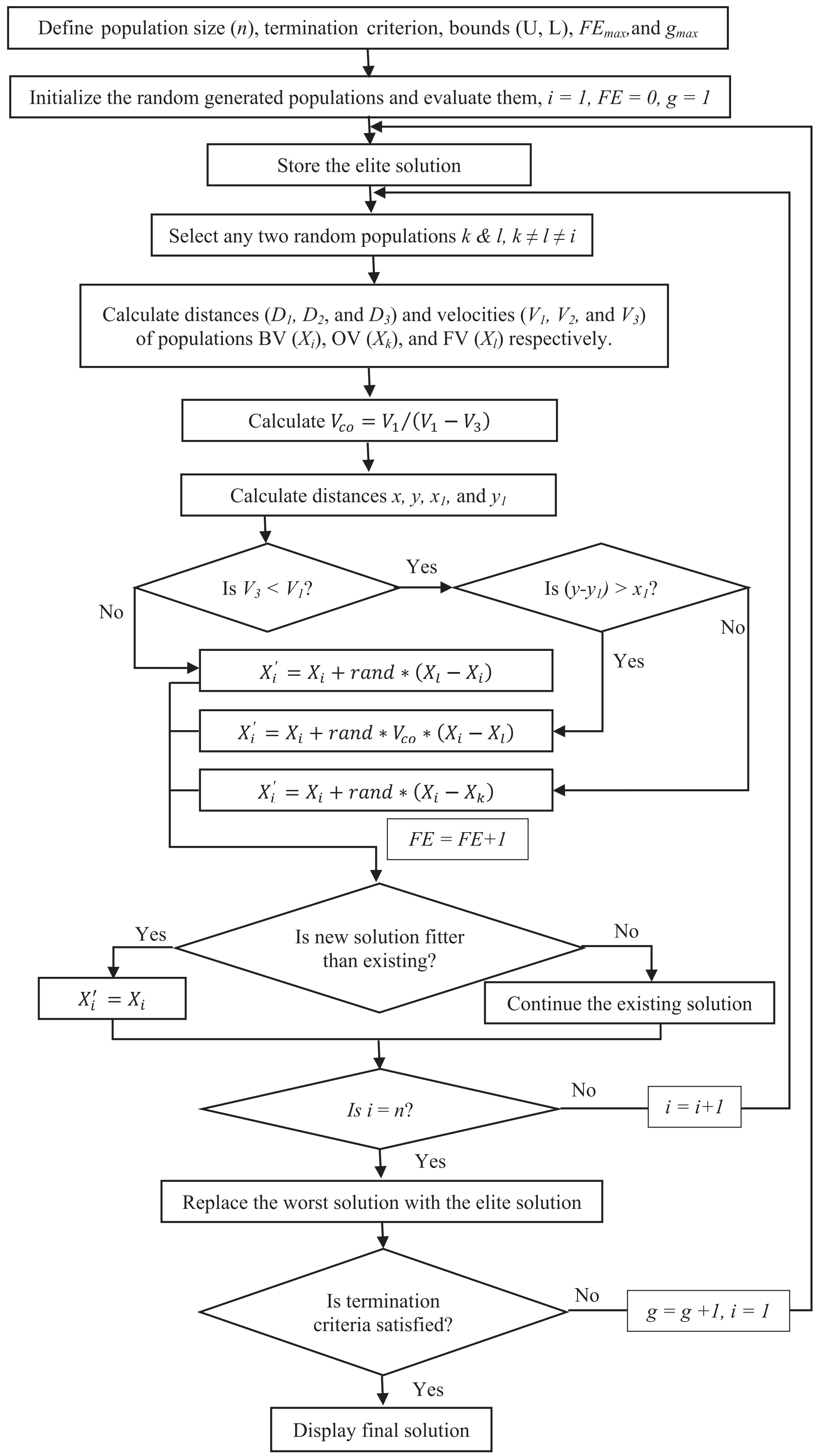

Fig. 7. Flow chart of the PVS algorithm.

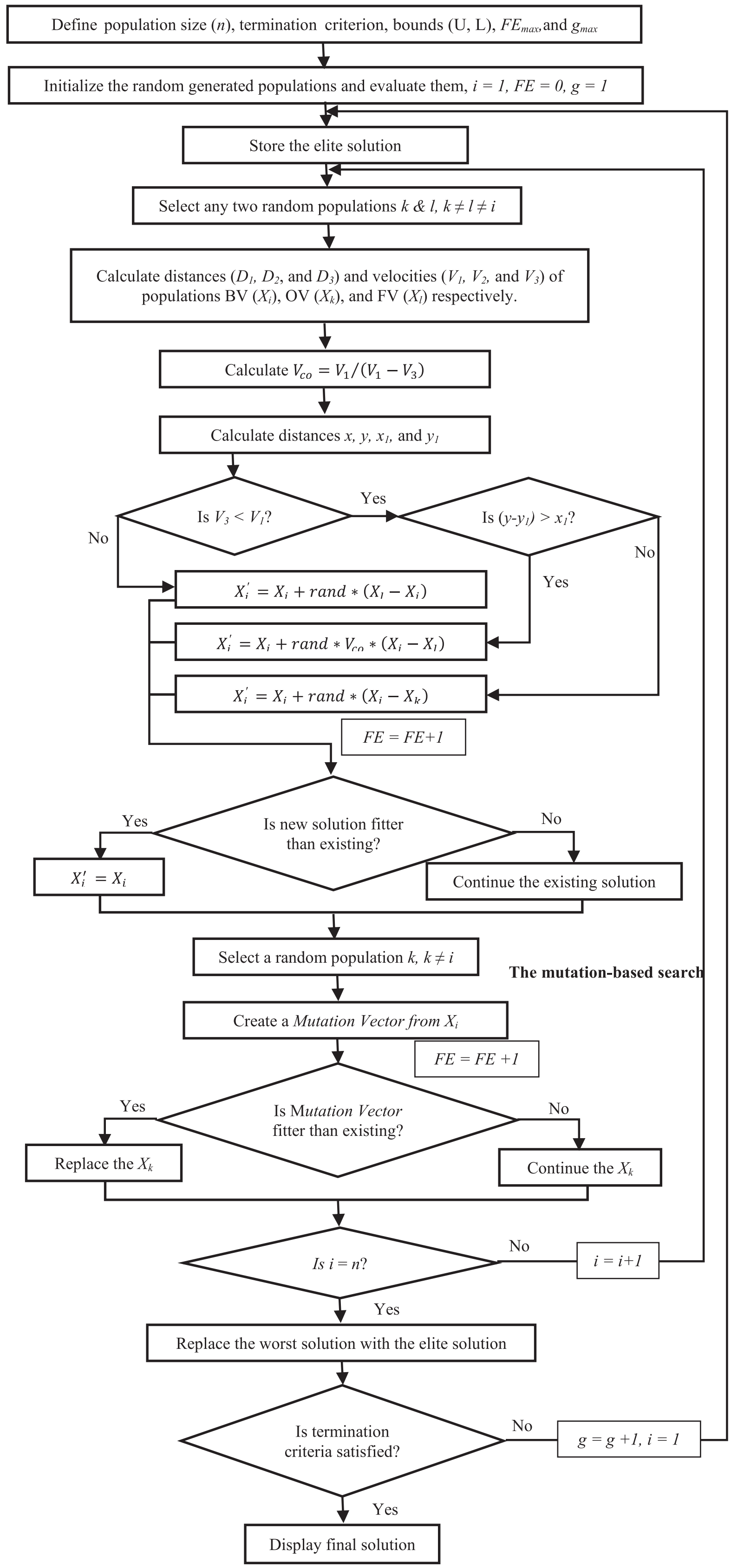

Fig. 8. Flow chart of the MPVS algorithm.

Please cite this article as: Savsani VJ, et al. Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Designand Engineering* (2016), http://dx.doi.org/10.1016/j.jcde.2016.10.002

generated populations. In addition, all the algorithms are coded with similar encoding platform. So, there is no change in the time complexity for the generation of initial solutions for basic and the modified versions of the algorithms. The time complexity for initial population/solution generation is $O(nd)$, where n indicate the size of the population and d is the dimension of the problem. Also, the time complexity for updating the solution for the basic and the modified algorithms also remains same as it uses the same solution updating formulas and it is also equal to $O(nd)$. However, the time complexity for the random mutation, which is incorporated to modify the algorithms is more than the basic algorithm by $O(nd)$. So, the modified algorithms increases the time complexity by $O(nd)$ compared to the basic variants of the algorithms. Total time consumed by the algorithm is the sum of all the time consumed by each part of the algorithms such as initial population generation, updating the solution and selection of updated solution and the random mutation (for modified algorithms).

b. Computational complexity: this type of complexity generally involves function evaluations and it is a function of population size and number of generations.

The proposed modification (the random mutation-based search) adds function evaluations equal to population size in each generation, which adds $O(n)$ computational complexity to the basic algorithm.

However, in this study, the algorithms' stopping criteria is controlled as per a maximum number of function evaluations, which means that algorithms stops at a particular number function evaluations set as a stopping criteria.

The function evaluations of the basic algorithms can be calculated as below:

- TLBO algorithm: function evaluations = 2 × population size (n) × number of generations(g) = 2 ng
- MTLBO algorithm: function evaluations = 3 × population size × number of generations = 3 ng
- HTS algorithm: function evaluations = population size × number of generations = ng
- MHTS algorithm: function evaluations = 2 × population size × number of generations = 2 ng
- PVS algorithm: function evaluations = population size × number of generations = ng
- MPVS algorithm: function evaluations = 2 × population size × number of generations = 2 ng
- WWO and MWWO: function evaluations are measured by using a function evaluation counter mechanism as it involves function evaluation depending on the execution of different solution updating phase, i.e., breaking, propagation and refraction.

It can be noted that the computational complexity is set identical for all the algorithms as function evaluations are used for the stopping criteria. So, the basic and the modified algorithms are stopped at same function evaluations. It is also noteworthy to note that the modified algorithms will consume less number of generations for the same function evaluations compared to the basic algorithms. This phenomenon is also reflected for the total time consumed by the basic and the modified algorithms. Total time consumed by the algorithms is the primary goal for the efficient algorithms; hence the basic and the modified algorithms are also compared for the total time by keeping fixed function evaluations.

4. Problem definition

The goal of the TTO is to minimize the weight of the truss by finding the best topology with optimum element cross-sectional areas such that it satisfies all stated constraints. Objective function considers element mass if an element exists and constant mass at each of the nodes if the node exists. Natural frequency, element stress, nodal displacement, Euler buckling, and kinematic stability constraints are considered in this investigation as discussed in the previous sections. Moreover, discrete cross-sectional areas are adopted to study practical structures. The mathematical formulation of the TTO problem can be performed as follows:

$$Find, \quad X = \{A_1, \quad A_2, .. \quad, A_n\} \tag{1}$$

$$that \quad minimize \quad the \quad weight \quad of \quad truss, \quad F(X) = \sum_{i=1}^{n} B_i A_i \rho_i L_i + \sum_{j=1}^{m} b_j$$

$$where \quad B_i = \begin{cases} 0, & if \quad A_i < Critical \quad area \\ 1; & if \quad A_i \geq Critical \quad area \end{cases}$$

and satisfies the constraints:

$$g_1(X): \quad Stress \quad constraints, \quad |B_i \sigma_i| - \sigma_i^{max} \leq 0$$

$$g_2(X): \quad Displacement \quad constraints, \quad |\delta_j| - \delta_j^{max} \leq 0$$

$$g_3(X): \quad Euler \quad buckling \quad constraints, \quad \left|B_i \sigma_i^{comp}\right| - \sigma_i^{cr} \leq 0, \quad where \quad \sigma_i^{cr} = \frac{k_i A_i E_i}{L_i^2}$$

$$g_4(X): \quad f_r - f_r^{max} \geq 0$$

$$g_5(X): \quad Cross-sectional \quad area \quad constraints, \quad A_i^{min} \leq A_i \leq A_i^{max}$$

$$g_6: \quad Check \quad on \quad validity \quad of \quad structure$$

$$g_7: \quad Check \quad on \quad kinematic \quad stability$$

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

where n and m are the numbers of elements and nodes of the truss respectively. The critical area is a small positive number [31]. If the cross-sectional area is smaller than the critical area, the element is assumed to be deleted from the truss. Otherwise, the element is retained in the truss. B_i is a topological bit, which is '0' for absence and '1' for the presence of the truss

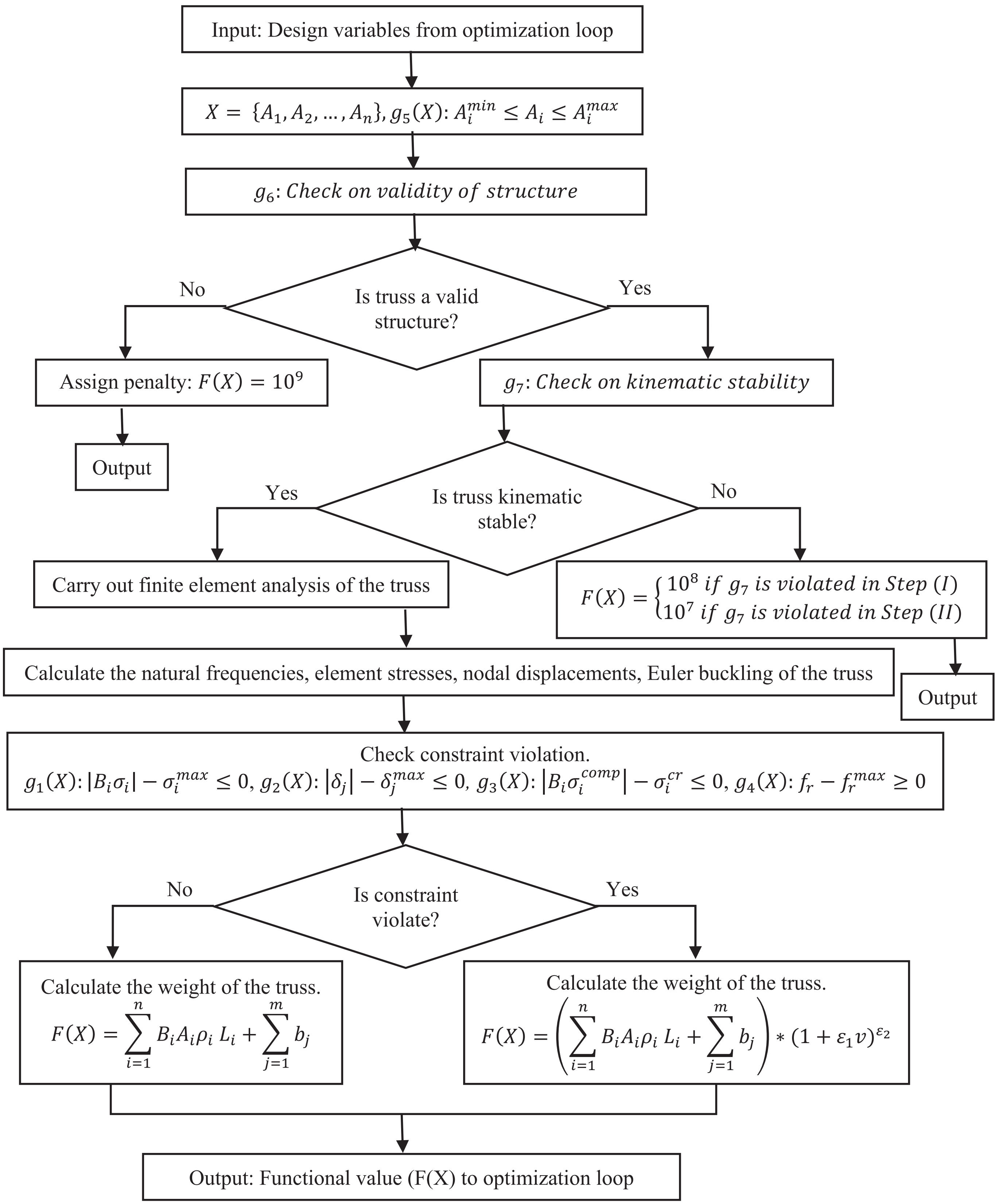

Fig. 9. Flow chart of the TTO problem.

ARTICLE IN PRESS

Table 1
The discrete sections [65].

Section number	Area (in.^{2})	Area (cm^2)	Mean diameter (cm)	Thickness (cm)	Section number	Area (in.^{2})	Area (cm^2)	Mean diameter (cm)	Thickness (cm)
1	0.111	0.7161	1.5098	0.1510	33	3.84	24.7741	8.8802	0.8880
2	0.141	0.9097	1.7017	0.1702	34	3.87	24.9677	8.9149	0.8915
3	0.196	1.2645	2.0062	0.2006	35	3.88	25.0322	8.9264	0.8926
4	0.25	1.6129	2.2658	0.2266	36	4.18	26.9677	9.2650	0.9265
5	0.307	1.9806	2.5109	0.2511	37	4.22	27.2258	9.3093	0.9309
6	0.391	2.5226	2.8337	0.2834	38	4.49	28.9677	9.6025	0.9602
7	0.442	2.8516	3.0128	0.3013	39	4.59	29.6128	9.7088	0.9709
8	0.563	3.6323	3.4003	0.3400	40	4.8	30.9677	9.9284	0.9928
9	0.602	3.8839	3.5161	0.3516	41	4.97	32.0645	10.1027	1.0103
10	0.766	4.9419	3.9662	0.3966	42	5.12	33.0322	10.2540	1.0254
11	0.785	5.0645	4.0151	0.4015	43	5.74	37.0322	10.8571	1.0857
12	0.994	6.4129	4.5181	0.4518	44	7.22	46.5806	12.1766	1.2177
13	1	6.4516	4.5317	0.4532	45	7.97	51.4193	12.7935	1.2793
14	1.228	7.9226	5.0218	0.5022	46	8.53	55.0322	13.2353	1.3235
15	1.266	8.1677	5.0989	0.5099	47	9.3	59.9999	13.8198	1.3820
16	1.457	9.4000	5.4700	0.5470	48	10.85	69.9999	14.9270	1.4927
17	1.563	10.0839	5.6655	0.5666	49	11.5	74.1943	15.3678	1.5368
18	1.62	10.4516	5.7679	0.5768	50	13.5	87.0966	16.6504	1.6650
19	1.8	11.6129	6.0799	0.6080	51	13.9	89.6772	16.8953	1.6895
20	1.99	12.8387	6.3927	0.6393	52	14.2	91.6127	17.0767	1.7077
21	2.13	13.7419	6.6138	0.6614	53	15.5	99.9998	17.8412	1.7841
22	2.38	15.3548	6.9911	0.6991	54	16	103.2256	18.1267	1.8127
23	2.62	16.9032	7.3352	0.7335	55	16.9	109.0320	18.6295	1.8630
24	2.63	16.9677	7.3491	0.7349	56	18.8	121.2901	19.6489	1.9649
25	2.88	18.5806	7.6905	0.7691	57	19.9	128.3868	20.2155	2.0216
26	2.93	18.9032	7.7570	0.7757	58	22	141.9352	21.2555	2.1255
27	3.09	19.9354	7.9660	0.7966	59	22.9	147.7416	21.6859	2.1686
28	3.13	20.1935	8.0174	0.8017	60	24.5	158.0642	22.4307	2.2431
29	3.38	21.8064	8.3314	0.8331	61	26.5	170.9674	23.3282	2.3328
30	3.47	22.3871	8.4416	0.8442	62	28	180.6448	23.9794	2.3979
31	3.55	22.9032	8.5383	0.8538	63	30	193.5480	24.8210	2.4821
32	3.63	23.4193	8.6340	0.8634	64	33.5	216.1286	26.2290	2.6229

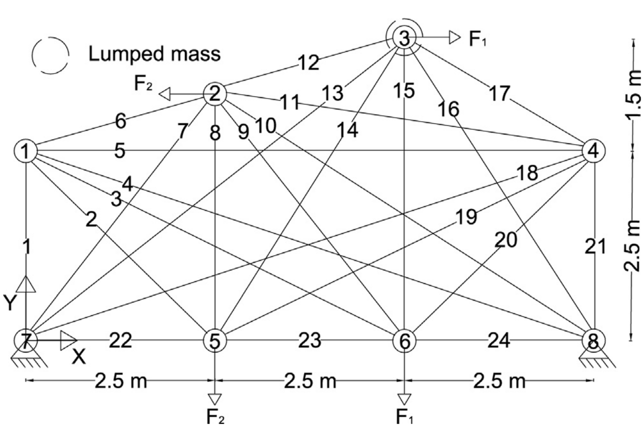

Fig. 10. Ground structure of the 24-bar truss.

element i. A_i, ρ_i, L_i, E_i, σ_i, and σ_i^{cr} correspond to the cross-sectional area, mass density, length, modules of elasticity, stress, and critical buckling of the element i respectively. δ_j and b_j are values of nodal displacement and weight of the node j respectively. f_r is rth natural frequency of the truss. The superscript, 'comp' corresponds to the compressive, whereas 'max' and 'min' correspond to the maximum and minimum allowable limits respectively. k_i is Euler buckling coefficient, which depends on the element cross-sectional geometry.

Truss structure is called invalid (g_6) if the truss has an absence of loaded node, support node, and non-erasable node ([55]).

In this study, kinematic stability (g_7) is tested in two steps as per [31] as explained below:

Step (I). Evaluate Grubler's criterion [62] to check degree of freedom of the truss:

$$\text{Degree of freedom} = d * m - n - m_r \quad (2)$$

where $d=2$ for planar truss and $d=3$ for space truss, n and m are the numbers of elements and nodes of the truss respectively, and m_r is restricted number of degrees of freedom at the support nodes. The generated truss must be kinematically stable so that it does not turn into a mechanism. If the degree of freedom is non-positive, the truss is not a mechanism. If the truss is a mechanism, we penalize the solution by assigning a large value. Thereafter, the truss is not sent to FEA model for further analysis.

Step (II). Evaluate positive definiteness of the global stiffness matrix (K) [63] to check singularity of the truss. Grubler's criterion, a necessary, yet not sufficient criterion for the kinematic stability. Therefore, a check of the positive-

ARTICLE IN PRESS

definiteness of the global stiffness matrix is still necessary [64]. Analytically, a truss is called kinematically unstable if the determinant of the stiffness matrix is zero; however, FEA model is a numerical method thus the determinant of the stiffness matrix may not be exact zero [23]. In this study, the command ‘eig(K)’ of MATLAB software is used to check positive definiteness of the global stiffness matrix. If the first the value of the ‘eig(K)’ is greater than 10^{-5} (a small number near to zero), the truss is assumed to be kinematic stable (positive-definite). If the truss is non-positive-definite, we penalize the solution by assigning a large value. Thereafter, the truss is not sent for further analysis to evaluate stresses, displacements, Euler buckling, and natural frequencies.

Penalty function approach is adapted to handle stated constraints. For no violation of the constraints, the penalty becomes zero; otherwise, the penalty is intended by following criteria [31,2]:

$$F(X)=\begin{cases} 10^9 \quad if\, g_6 \quad is \quad violated \\ 10^8 \quad if\, g_7 \quad is \quad violated \quad in \quad Step \quad (I) \\ 10^7 \quad if\, g_7 \quad is \quad violated \quad in \quad Step \quad (II) \\ F(X)*F_{penalty}(X) \quad otherwise \end{cases} \tag{3}$$

$$where, \quad F_{penalty}(X)=(1+\varepsilon_1 v)^{\varepsilon_2}, \quad v=\sum_{i=1}^{q}\left|\frac{1-p_i}{p_i^*}\right| \tag{4}$$

where p_i signifies the value of constraint violation of ith constraint, p_i^* is its bound, and q is an active number of constraints. The parameters ε_1 and ε_2 are selected in view of the constraint violation. Values of ε_1 and ε_2 are set as 2 in this study by investigating its effect and referring to the previous studies. The detailed graphical representation of the formulation of the TTO problem is illustrated in Fig. 9.

Table 2
Design considerations of the 24-bar truss.

Design variables: A_i, $i=1,2,\ldots,24$		
Loading condition 1:	Loading condition 2:	
$F_1=100$ kN; $F_2=0$ kN	$F_1=0$ kN; $F_2=100$ kN	
$\sigma_i^{max}=180$ MPa	$\delta_{5y,6y}^{max}=10$ mm	$f_1 \geq 30$ Hz
$E=200$ GPa	$\rho=7860$ kg/m^3	

5. Design problems and discussions

In this section, three distinct benchmark problems [1-5] are introduced with discrete design parameters to answer practicability of truss structures. In all of the truss problems, the cross-sections are assumed to be tubular with a ratio of mean diameter to wall thickness of approximately 10. Euler buckling

Table 3
Optimal discrete design parameters for the 24-bar truss (weight does not consider lumped mass).

Variable	TLBO	MTLBO	HTS	MHTS	WWO	MWWO	PVS	MPVS
A_1	–	–	–	–	0.9097	–	–	–
A_2	0.7161	–	–	–	–	–	–	–
A_5	0.7161	–	–	–	–	–	–	–
A_6	–	–	–	–	0.9097	–	–	–
A_7	18.5806	18.5806	18.5806	18.5806	18.5806	18.5806	18.9032	18.5806
A_8	6.4516	6.4129	6.4129	6.4516	10.0839	6.4129	6.4129	6.4516
A_9	6.4129	3.8839	–	1.9806	–	6.4129	1.2645	1.6129
A_{10}	–	–	1.2645	–	–	–	–	–
A_{11}	–	–	–	–	1.2645	–	–	–
A_{12}	3.8839	2.5226	–	2.5226	1.6129	3.6323	2.5226	2.5226
A_{13}	–	–	6.4129	–	–	–	–	–
A_{14}	–	1.2645	–	2.5226	7.9226	–	2.8516	2.5226
A_{15}	3.6323	4.9419	6.4129	6.4129	8.1677	3.6323	6.4129	6.4129
A_{16}	–	–	19.9354	19.9354	19.9354	–	19.9354	19.9354
A_{17}	10.0839	10.0839	–	–	0.9097	10.0839	–	–
A_{19}	0.7161	–	–	–	–	–	–	–
A_{20}	6.4129	6.4516	–	–	–	6.4129	–	–
A_{21}	10.4516	10.0839	–	–	–	10.0839	–	–
A_{22}	–	0.9097	–	1.6129	–	0.7161	1.6129	1.6129
A_{23}	–	–	0.7161	–	3.6323	–	–	–
A_{24}	8.1677	7.9226	1.2645	0.9097	2.8516	8.1677	0.9097	1.2645
Weight (kg)	243.2922	227.3884	242.4321	226.3047	281.4191	229.0035	226.1573	225.8168
σ_{max} (MPa)	165.3950	164.4027	155.9359	155.2136	122.4330	165.3950	158.8640	155.3037
σ_{max}^{cr} (MPa)	133.7804	133.7803	88.4133	88.4133	88.4133	129.0733	89.9483	88.4133
f_1 (Hz)	30.1568	30.0719	30.0459	30.0763	30.0047	30.2718	30.0048	30.0152
δ_{5y} (mm)	3.2037	2.8467	1.1755	2.3577	0.9721	3.2142	2.1874	2.3277
δ_{6y} (mm)	3.7693	3.4258	3.5820	3.3273	2.9524	3.7777	3.3623	3.3282
Mean	336.9759	283.1211	394.8635	300.9344	478.4674	369.6346	282.4255	280.7890
SD	86.8971	53.7031	125.5721	70.2199	207.2567	75.5695	57.1218	47.4631
Mean time (s)	25.14	19.27	36.18	25.38	17.28	16.71	23.24	19.33

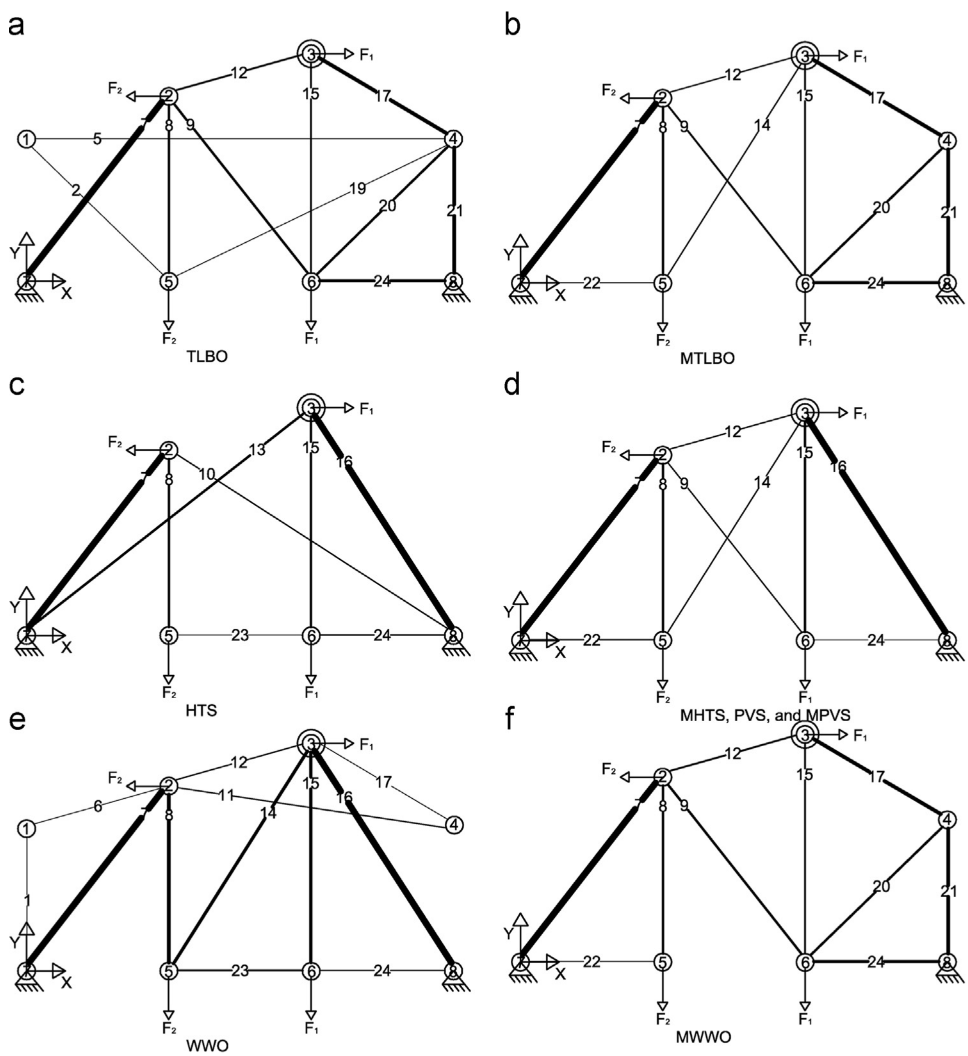

Fig. 11. The 24-bar truss.

coefficient (k_i, $i=1,2,\ldots,n$) and mass of each of the nodes (b_j, $j=1,2,.,m$) are assumed to be 4.0 and 5 kg respectively for all problems [1,2,6]. The discrete design variables may take any integer values within [–64, 64], where positive integer denotes element cross-sectional number [65], illustrated in Table 1, and zero or a negative integer of design variable signifies removal of the element. Moreover, the search space is nearly converted into two times of design variable limits in order to reveal topology optimization as described before. As noted in the previous sections, all the problems consider stress, displacement, buckling, and frequency as constraints along with multi-load conditions.

The benchmark problems with the discrete cross-sectional area are introduced the first time in this study. Therefore, no experimental investigation is observed for these problems in the literature. Thus, comparison among the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms are considered in this study. The considered algorithms are measured with 100 independent runs for each problem to study the stochastic nature of the meta-heuristics. All problems are investigated by taking a population size of 50. Whereas, the large size truss (72-bar truss) is examined for the population size of 100. The TLBO, MTLBO, HTS, MHTS, PVS, and MPVS algorithms do not need other algorithm controlling parameters, whereas the WWO and MWWO algorithms consider these parameters as the maximum wave height $h_{max}=12$, the wavelength reduction coefficient $\alpha=1.0026$, the breaking coefficients $\beta_{max}=0.25$, $\beta_{min}=0.001$, and the maximum number of breaking directions $k_{max}=12$. The results and discussion are explained in the subsequent sections:

5.1. A 24-bar 2-D truss

The first benchmark problem for the 24-bar planar truss is considered to investigate the influence of discrete design variables, shown in Table 1. This problem was investigated in [1-5] by considering continuous design variables; however, this study investigates it for discrete design variables. The ground structure of this truss is presented in Fig. 10. Design considerations such as multi-load conditions, constraints, and material properties are tabulated in Table 2. The truss is subjected to a non-structural lumped mass of 500 kg at node 3.

In this section, the proposed algorithms are investigated on discrete TTO by considering a population size and function evaluations (FE_{max}) as 50 and 20,000 respectively. The results are obtained for 100 independent runs and the statistical results obtained in these runs are presented in Table 3. The results show that the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give trusses with the best weight of 243.2922, 227.3884, 242.4321, 226.3047, 281.4191, 229.0035, 226.1573, and 225.8168 kg respectively. The results signify that the MPVS algorithm ranks first among all considered meta-heuristics, whereas the PVS and MHTS stand second and third respectively. Thus, the weight benefit for the MPVS algorithm is 17.4754, 1.5716, 16.6153, 0.4879, 55.6023, 3.1867, and 0.3405 kg as compared to those obtained from the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, and PVS algorithms respectively. The results show that the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give the mean weight of 336.9759, 283.1211, 394.8635, 300.9344, 478.4674, 369.6346, 282.4255, and 280.7890 kg respectively. Therefore, the MPVS algorithm performs best among all considered algorithms to obtain the minimum mean weight. The mean weight benefit for the MPVS algorithm is 56.1869, 2.3321, 114.0745, 20.1454, 197.6784, 88.8456, and 1.6365 kg as compared to those obtained from the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, and PVS algorithms respectively. Whereas the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give the standard deviation (SD) of weight as 86.8971, 53.7031, 125.5721, 70.2199, 207.2567, 75.5695, 57.1218, and 47.4631 respectively. It can be seen from the results that the MPVS algorithm performs better in order to get the best weight, mean weight, and SD of weight. Fig. 11 presents the relative virtual effect on element cross-sectional areas and best topologies on each truss obtained by various approaches. It is also observed from the figure that the MHTS, PVS, and MPVS algorithms set identical topologies. The figure also shows that best topologies obtained using the

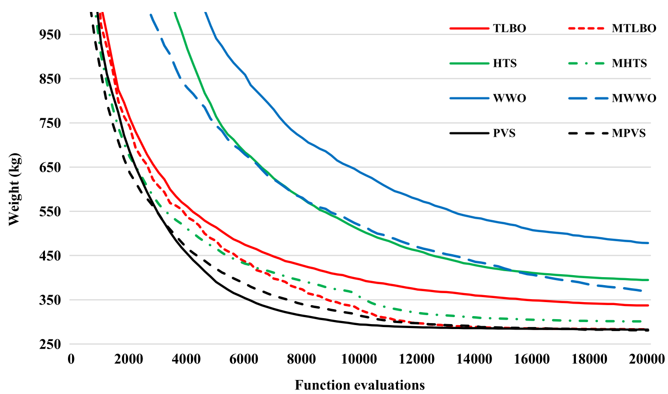

Fig. 12. Convergence graph of the 24-bar truss.

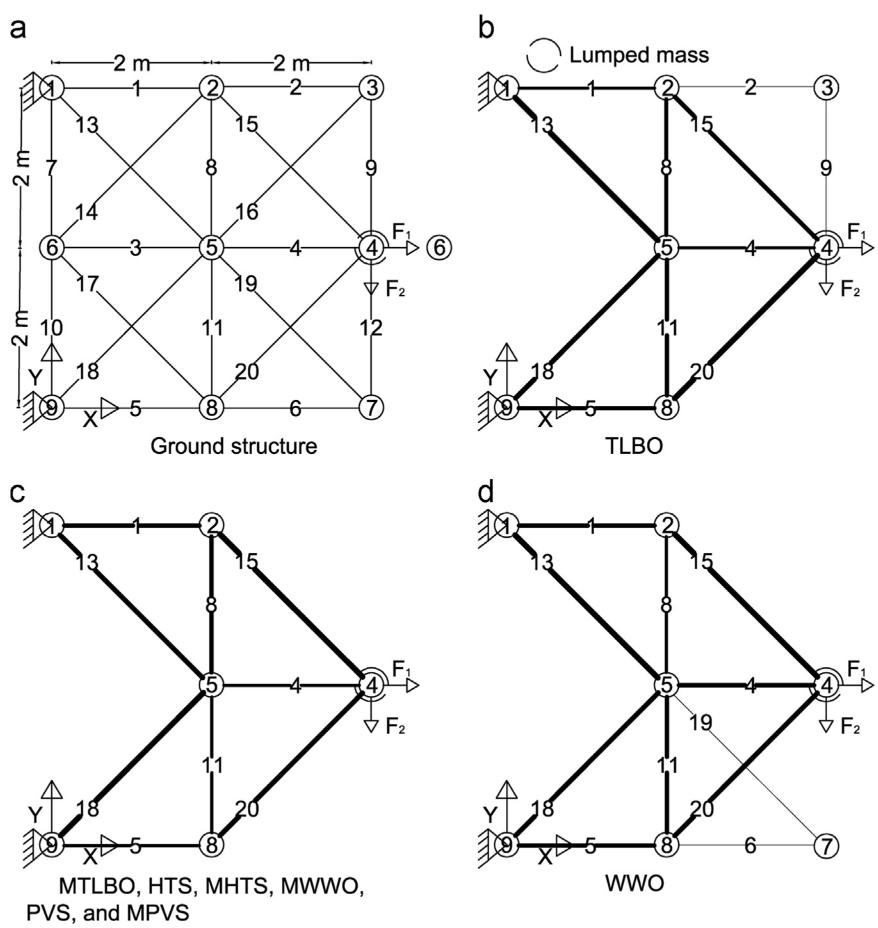

Fig. 13. The 20-bar truss.

MHTS, PVS, and MPVS algorithms need only 9 elements, whereas the HTS algorithm consumes only 8 elements out of 24 of the ground structure. Moreover, topologies obtained using the MHTS, PVS, and MPVS algorithms offer lighter weight among the considered algorithms, while the HTS algorithm designs heavier truss. Therefore, the results suggest that the optimum distribution of elements plays a significant role in the TTO. In addition, a number of elements in the truss do not have a major impact in order to obtain lighter truss.

Fig. 12 presents convergence graph of mean weight obtained by the proposed algorithms of the test problem. The mean weight is computed by considering the average weight of all runs for each generation. The convergence graph indicates that the MTLBO, PVS, and MPVS algorithms converge faster and set superior solutions as compared to other algorithms. Moreover, the PVS algorithm shows early convergence nearly within 14,000 FE, but the MPVS algorithm sets the best convergence. Moreover, the MTLBO, MHTS, and MWWO algorithms have good convergence performance in comparison with their basic versions.

The algorithms are also compared based on the total time required for the intended function evaluations. Results reveal that the modified algorithms require less total time to complete the required function evaluations. Therefore, it is observed from the results and the convergence graph that the performance of the proposed modified algorithms is better as compared to their basic versions.

5.2. A 20-bar 2-D truss

The ground structure of the second benchmark problem is illustrated in Fig. 13. This problem was optimized in [1-5] for continuous optimization, whereas this study investigates discrete optimization. The discrete element cross-sections, shown in Table 1. Design considerations such as multi-load conditions, constraints, and material properties are presented in Table 4. The truss is subjected to non-structural lumped mass of 200 kg at node 4.

Table 4
Design considerations of the 20-bar truss.

Design variables: A_i, $i=1,2,\ldots,20$	
Loading condition 1:	Loading condition 2:
$F_1=500$ kN; $F_2=0$ kN	$F_1=0$ kN; $F_2=500$ kN
$\sigma_i^{max}=180$ MPa $\delta_{4y}^{max}=10$ mm	$f_1 \geq 60$ Hz $f_2 \geq 100$ Hz
$E=200$ GPa	$\rho=7860$ kg/m^3

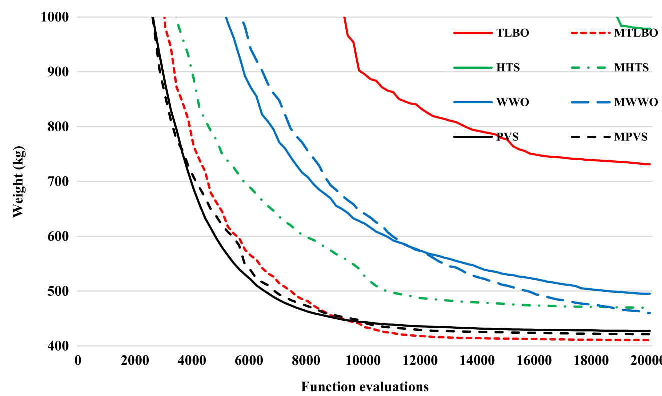

Fig. 14. Convergence graph of the 20-bar truss.

Table 5
Optimal discrete design parameters for the 20-bar truss (weight does not consider lumped masses).

Variable	TLBO	MTLBO	HTS	MHTS	WWO	MWWO	PVS	MPVS
A_1	13.7419	13.7419	13.7419	15.3548	16.9032	20.1935	16.9032	15.3548
A_2	0.7161	–	–	–	–	–	–	–
A_4	13.7419	13.7419	12.8387	13.7419	16.9677	16.9032	13.7419	13.7419
A_5	16.9677	16.9032	15.3548	16.9032	16.9032	16.9032	13.7419	15.3548
A_6	–	–	–	–	0.9097	–	–	–
A_8	15.3548	13.7419	16.9677	13.7419	13.7419	22.3871	16.9032	15.3548
A_9	0.7161	–	–	–	–	–	–	–
A_{11}	16.9032	16.9032	18.5806	16.9677	16.9677	12.8387	13.7419	15.3548
A_{13}	22.9032	21.8064	19.9354	22.3871	22.9032	18.5806	19.9354	21.8064
A_{15}	18.5806	19.9354	21.8064	18.5806	22.3871	21.8064	21.8064	19.9354
A_{18}	19.9354	20.1935	19.9354	18.9032	20.1935	22.3871	22.3871	19.9354
A_{19}	–	–	–	–	1.9806	–	–	–
A_{20}	21.8064	22.3871	21.8064	23.4193	20.1935	18.9032	20.1935	22.3871
Weight (kg)	342.8615	335.4113	337.4007	335.7534	359.3985	351.8429	335.4113	335.0404
σ_{max} (MPa)	179.7263	177.1362	179.8252	177.3407	178.9374	177.8930	177.6526	176.2678
σ_{max}^{cr} (MPa)	339.3542	338.0638	371.6122	339.3542	339.3542	447.7410	338.0638	307.0962
f_1 (Hz)	64.9482	65.3759	65.1849	65.2803	65.2876	65.6465	65.4716	65.3335
f_2 (Hz)	114.3366	114.3824	113.5731	114.3286	106.4782	116.9238	114.4201	114.4593
δ_{4y} (mm)	9.9991	9.9909	9.9209	9.9986	9.7465	9.7684	9.9909	9.9951
Mean	731.3855	410.4343	978.7233	469.7096	495.1262	459.6531	427.3825	421.3447
SD	903.2378	52.8557	1488.4232	269.3849	145.9844	60.0836	59.2259	53.7251
Mean time (s)	28.76	22.56	37.82	26.91	11.82	11.76	27.97	22.51

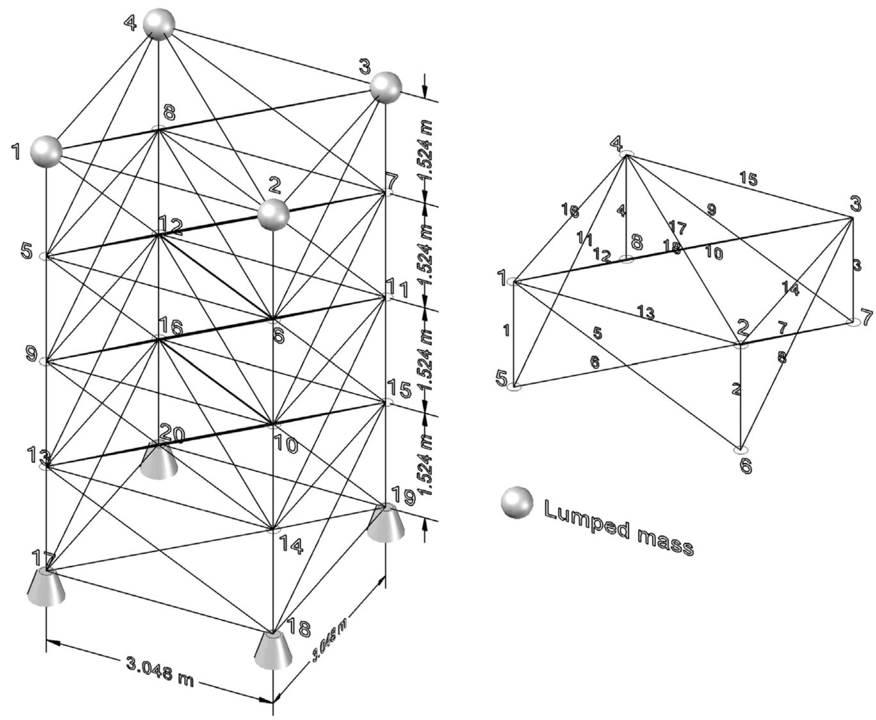

Fig. 15. The 72-bar truss.

Table 5 shows the results obtained by using the proposed algorithms. The best weights for the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms are obtained as 342.8615, 335.4113, 337.4007, 335.7534, 359.3985, 351.8429, 335.4113, and 335.0404 kg respectively. The results show that the MPVS algorithm ranks first among considered meta-heuristics, whereas the PVS and MTLBO algorithms stand second. Thus, the weight benefit for the MPVS algorithms is 7.8211, 0.3709, 2.3603, 0.7130, 24.3581, 16.8025, and 0.3709 kg as compared to those obtained from the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, and PVS algorithms respectively. The results show that the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give the mean weight of 731.3855, 410.4343, 978.7233, 469.7096, 495.1262, 459.6531, 427.3825, and 421.3447 kg respectively. Therefore, the MTLBO algorithm performs best among considered algorithms to obtain minimum mean weight. The mean weight benefit for the MTLBO algorithm is 320.9512, 568.2890, 59.2753, 84.6919, 49.2188, 16.9482, and 10.9104 kg as compared to those obtained from the TLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms respectively. Whereas the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give the SD of weight as 903.2378, 52.8557, 1488.4232, 269.3849, 145.9844, 60.0836, 59.2259, and 53.7251 respectively. Therefore the MTLBO algorithm performs better in order to get better mean and SD of weight as the truss. Fig. 13 also demonstrates the relative virtual effect on element cross-sectional areas and best topologies of each of the truss obtained by using various approaches. It is also observed from the figure that the MTLBO, HTS, MHTS, MWWO, PVS, and MPVS algorithms give identical topologies. Moreover, the MTLBO, HTS, MHTS, MWWO, PVS, and MPVS algorithms design lighter truss with 8 elements out of 20 in the ground structure, which clarifies that the identification of the optimum topology is the most important factor in the TTO.

Table 6
Design considerations of the 72-bar truss.

Design variables: G_i, $i=1,2,\ldots,16$, where G is group number			
Loading condition 1:		Loading condition 2:	
$F_{1x}=F_{1y}=22.25$ kN;		$F_{1z}=F_{2z}=F_{3z}=F_{4z}=-22.25$ kN	
$F_{1z}=-22.25$ kN			
$\sigma_i^{max}=180$ MPa		$\delta_j^{max}=6.35$ mm *(for nodes, j=1, 2, 3 and 4 along x- and y-axes)*	
$f_1 \geq 4$ Hz	$f_3 \geq 6$ Hz	$E=200$ GPa	$\rho=7860$ kg/m^3

Fig. 14 presents convergence graph of the mean weight for the proposed algorithms of the test problem. The convergence graph indicates that the MTLBO and MPVS algorithms converge faster and achieves good optimal results as compared to other algorithms. Moreover, the PVS algorithm shows early convergence nearly within 10,000 FE, but the MPVS algorithm sets the better results. It is also observed from the convergence graph that the MTLBO, MHTS, and MWWO algorithm have good convergence performance in comparison with their basic versions. Therefore, the performance of the proposed modified algorithms is better as compared to their basic versions in terms of statistical results, convergence and total time.

Table 7
Optimal discrete design parameters for the 72-bar truss (weight does not consider lumped masses).

Variable	TLBO	MTLBO	HTS	MHTS	WWO	MWWO	PVS	MPVS
G_1	2.8516	2.8516	2.8516	3.6323	2.8516	2.5226	2.8516	2.8516
G_2	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323
G_3	2.8516	2.8516	2.8516	2.8516	2.8516	2.8516	2.8516	2.8516
G_4	5.0645	5.0645	5.0645	5.0645	6.4129	5.0645	5.0645	5.0645
G_5	2.8516	2.8516	4.9419	2.8516	3.6323	3.6323	2.8516	2.8516
G_6	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323
G_9	4.9419	4.9419	3.8839	4.9419	3.6323	4.9419	4.9419	4.9419
G_{10}	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323	3.6323
G_{13}	3.8839	3.8839	3.8839	3.8839	4.9419	4.9419	3.8839	3.8839
G_{14}	3.6323	3.6323	3.6323	3.6323	3.8839	3.6323	3.6323	3.6323
Weight (kg)	542.5877	542.5877	547.5337	546.3281	559.6509	549.8213	542.5877	542.5877
σ_{max} (MPa)	80.8928	80.8928	82.2067	81.0390	76.4686	86.7873	80.8928	80.8928
σ_{max}^{cr} (MPa)	170.2224	170.2224	170.2224	170.2224	170.2224	170.2224	170.2224	170.2224
f_1 (Hz)	4.0197	4.0197	4.0206	4.0297	4.1214	4.2390	4.0197	4.0197
f_3 (Hz)	6.8520	6.8520	6.8508	6.8509	6.9073	6.8518	6.8520	6.8520
δ_{max} (mm)	2.9086	2.9086	2.9669	2.9173	2.7717	2.6914	2.9086	2.9086
Mean	684.4426	676.2892	888.1696	690.8083	815.9453	813.7952	648.2609	629.7766
SD	218.0771	73.3845	417.6501	73.3499	240.9534	78.2745	68.2537	68.1843
Mean time (s)	99.59	50.86	121.57	74.09	38.99	33.73	91.58	55.44

5.3. A 72-bar 3-D truss

The ground structure of the third benchmark problem is illustrated in Fig. 15. This problem was recently optimized in [2,4,5] for continuous optimization. However, this study investigates this problem for discrete optimization. The discrete element cross-sections are selected, shown in Table 1. Design considerations such as continuous design variables, multi-load conditions, constraints, material properties, and truss geometry data are summarized in Table 6. The elements are clustered into 16 groups (i.e., G_1 (A_1–A_4), G_2 (A_5–A_{12}), G_3 (A_{13}–A_{16}), G_4 (A_{17}–A_{18}), G_5 (A_{19}–A_{22}), G_6 (A_{23}–A_{30}), G_7 (A_{31}–A_{34}), G_8 (A_{35}–A_{36}), G_9 (A_{37}–A_{40}), G_{10} (A_{41}–A_{48}), G_{11} (A_{49}–A_{52}), G_{12} (A_{53}–A_{54}), G_{13} (A_{55}–A_{58}), G_{14} (A_{59}–A_{66}), G_{15} (A_{67}–A_{70}), and G_{16} (A_{71}–A_{72})) by considering structural symmetry as per [2]. The figure also states element connectivity of elements A_1 to A_{18} and the rest of elements are in the similar pattern (i.e., $A_{(i+18i)}$ to $A_{(4i+18i)}$) for $i=1,2,3$, and 4). The truss is subjected to four non-structural lumped masses of 2270 kg at each of the top nodes (nodes 1–4).

Table 7 compares the optimum results for 100 independent runs obtained from this work. The results indicate that the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms design optimum truss with minimum weight 542.5877, 542.5877, 547.5337, 546.3281, 559.6509, 549.8213, 542.5877, 542.5877 kg respectively. It can be seen from the results that the TLBO, MTLBO, PVS, and MPVS algorithms give identical trusses and ranks first among considered meta-heuristics in order to achieve light weight truss. Thus, the weight benefit for the TLBO, MTLBO, PVS, and MPVS algorithms is 4.9460, 3.7404, 17.0632, and 7.2336 kg as compared to those obtained from the HTS, MHTS, WWO, and MWWO algorithms respectively. The results show that the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms provide the mean weight of 684.4426, 676.2892, 888.1696, 690.8083, 815.9453, 813.7952, 648.2609, and 629.7766 kg respectively. The results signify that the MPVS algorithm ranks first among considered meta-heuristics, whereas the PVS and MTLBO algorithms stand second and third respectively. Thus, the mean weight benefit for the PVS algorithm is 54.6660, 46.5126, 258.3930, 61.0317, 186.1687, 184.0186, and 18.4843 kg as compared to those obtained from the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, and PVS algorithms respectively. Whereas the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms give the SD of weight as 218.0771, 73.3845, 417.6501, 73.3499, 240.9534, 78.2745, 68.2537, and 68.1843 respectively. The results indicate that the MPVS algorithm performs better in order to get better SD of weight as the truss.

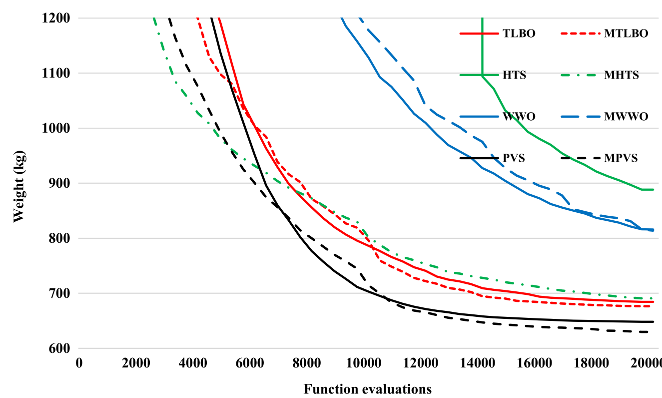

Fig. 16. Convergence graph of the 72-bar truss.

Fig. 16 shows convergence graph of the mean weight for the proposed algorithms of the 72-bar truss. The convergence graph indicates that the MPVS algorithm shows the fastest convergence and set better results among considered algorithm. Moreover, the PVS and MPVS algorithms show better

Table 8
Result summary.

	Weight (kg)	TLBO	MTLBO	HTS	MHTS	WWO	MWWO	PVS	MPVS
24-bar truss	Minimum	243.2922	227.3884	242.4321	226.3047	281.4191	229.0035	226.1573	225.8168
	Mean	336.9759	283.1211	394.8635	300.9344	478.4674	369.6346	282.4255	280.789
	SD	86.8971	53.7031	125.5721	70.2199	207.2567	75.5695	57.1218	47.4631
20-bar truss	Minimum	342.8615	335.4113	337.4007	335.7534	359.3985	351.8429	335.4113	335.0404
	Mean	731.3855	410.4343	978.7233	469.7096	495.1262	459.6531	427.3825	421.3447
	SD	903.2378	52.8557	1488.4232	269.3849	145.9844	60.0836	59.2259	53.7251
72-bar truss	Minimum	542.5877	542.5877	547.5337	546.3281	559.6509	549.8213	542.5877	542.5877
	Mean	684.4426	676.2892	888.1696	690.8083	815.9453	813.7952	648.2609	629.7766
	SD	218.0771	73.3845	417.6501	73.3499	240.9534	78.2745	68.2537	68.1843

Table 9
The Friedman rank test of minimum, mean, and SD of weight.

Algorithms	Test for the minimum solution			Test for the mean solution			Test for the SD		
	Friedman value	Normalized value	Rank	Friedman value	Normalized value	Rank	Friedman value	Normalized value	Rank
TLBO	15.5	0.3483	5	16	0.3810	5	19	0.4419	6
MTLBO	9	0.2022	3	7	0.1667	2	7	0.1628	2
HTS	17	0.3820	6	23	0.5476	8	23	0.5349	8
MHTS	12	0.2697	4	14	0.3333	4	13	0.3023	4
WWO	24	0.5393	8	21	0.5000	7	20	0.4651	7
MWWO	19	0.4270	7	16	0.3810	5	14	0.3256	5
PVS	7	0.1573	2	7	0.1667	2	8	0.1860	3
MPVS	4.5	0.1011	1	4	0.0952	1	4	0.0930	1

convergence as compared to other considered meta-heuristics. It is also observed from the convergence graph that the MTLBO, MHTS, and MWWO algorithm have good convergence performance in comparison with their basic versions. Therefore, the performance of the proposed modified algorithms is better as compared to their basic versions.

Result summary of the proposed algorithms is presented in Table 8. It can be understood from the summary table the MPVS algorithm ranks first in the 24-bar truss and 20-bar truss, whereas the TLBO, MTLBO, PVS, and MPVS algorithms give ideal solutions and rank first in the 72-bar truss in order to get minimum weight. The MPVS algorithm is better in the 24-bar truss and 72-bar truss, whereas the MTLBO algorithm ranks first in the 20-bar truss in order to provide better mean weight. Moreover, the MPVS algorithm gives best SD of weight for the 24-bar truss, 20-bar truss, and 72-bar truss. Therefore, the results signify that the MPVS algorithm performs best, followed by the MTLBO algorithm among the proposed algorithms. Furthermore, it is also observed from the result tables that modified algorithms require less computational time than their basic versions.

From the results, it can be understood that the MPVS algorithm is a better-performing algorithm, but at the same time, statistical tests are also important to give rank to all the algorithms based on the obtained results by the proposed method over other comparative algorithms [66]. Therefore, the Friedman rank test is performed at the minimum, mean, SD of weight obtained by the TLBO, MTLBO, HTS, MHTS, WWO, MWWO, PVS, and MPVS algorithms respectively. Table 9 presents the Friedman rank test for the test problems. The results of the Friedman test are normalized with respect to the best value obtained, and algorithms are ranked based on the normalized value. The results indicate that the MPVS algorithm stands first, whereas the PVS and MTLBO algorithms rank second and third respectively to obtain minimum weight. The MPVS algorithm ranks first, whereas the PVS and MTLBO algorithms stand second to obtain better mean weight respectively. The MPVS, MTLBO, and PVS algorithms rank first, second, and third to obtain better SD of weight respectively. It is also observed from the results that the proposed modification has improved the performance of each of the considered algorithms significantly.

6. Conclusions

In this paper, the four basic meta-heuristics (viz., TLBO, HTS, WWO, and PVS) and four modified meta-heuristics (viz., MTLBO, MHTS, MWWO, and MPVS) are proposed for the TTO to design planar and space trusses with static and dynamic constraints using single stage solution approach. The discrete variables are introduced in order to consider manufacturability. The proposed algorithms are applied successfully on three benchmark problems of the TTO to investigate their performance. All benchmark problems are examined by

Please cite this article as: Savsani VJ, et al. Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Designand Engineering* (2016), http://dx.doi.org/10.1016/j.jcde.2016.10.002

considering, stress, displacement, buckling, frequency, and kinematic stability constraints. The simultaneous attention of the stated constraints makes the TTO problems complex and challenging, which demands effective optimization algorithms to achieve the best solution. The TTO works on the removal of superfluous elements and nodes, which ends up with a great saving in the weight and this advantage is raised up by considering the large nodal weight. In this paper, FEA model is revised, which results in a restructuring of stiffness, mass, and load matrix during the course of optimization, to avoid singularity and unnecessary analysis.

The results of the Friedman rank test show that the MPVS algorithm ranks first in order to achieve lighter trusses, followed by the PVS and MTLBO algorithms. Moreover, the MPVS algorithm ranks first to obtain better mean and SD of weight respectively. The results of the MPVS algorithm are observed superior and more reliable as compared other results of the proposed algorithms. Moreover, the random mutation-based search has enhanced the exploration and exploitation capacities of the basic TLBO, HTS, WWO, and PVS algorithms.

Conflict of interest

All the authors agree that they do not have any conflict of interest with anyone for the submitted manuscript.

References

[1] Xu B, Jiang J, Tong W, Wu K. Topology group concept for truss topology optimization with frequency constraints. *J. Sound Vib.* 2003;**261**(5)911–25.

[2] Kaveh A, Zolghadr A. Topology optimization of trusses considering static and dynamic constraints using the CSS. *Appl. Soft Comput. J.* 2013;**13**(5)2727–34.

[3] Gonçalves MS, Lopez RH, Miguel LFF. Search group algorithm: a new metaheuristic method for the optimization of truss structures. *Comput. Struct.* 2015;**153**:165–84.

[4] Kaveh A, Mahdavi VR. Colliding bodies optimization for size and topology optimization of truss structures. *Struct. Eng. Mech.* 2015;**53**(5)847–65.

[5] Savsani VJ, Tejani GG, Patel VK. Truss Topology Optimization (TTO) with static and dynamic constraints using the modified sub-population teaching learning based optimization (MS-TLBO). *Eng. Optim.* 2016. http://dx.doi.org/10.1080/0305215X.2016.1150468.

[6] Tejani GG, Savsani VJ, Patel VK. Adaptive Symbiotic Organisms Search (SOS) algorithm for structural design optimization. *J. Comput. Des. Eng.* 2016. http://dx.doi.org/10.1016/j.jcde.2016.02.003.

[7] Tejani GG, Savsani VJ, Patel VK. Modified sub-population teaching-learning-based optimization for design of truss structures with natural frequency constraints. *Mech. Based Des. Struct. Mach.* 2016. http://dx.doi.org/10.1080/15397734.2015.1124023.

[8] Hajirasouliha I, Pilakoutas K, Moghaddam H. Topology optimization for the seismic design of truss-like structures. *Comput. Struct.* 2011;**89**:702–11. http://dx.doi.org/10.1016/j.compstruc.2011.02.003.

[9] Cheng MY, Prayogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput. Struct.* 2014;**139**:98–112.

[10] Ohsaki M, Fujisawa K, Katoh N, Kanno Y. Semi-definite programming for topology optimization of trusses under multiple eigenvalue constraints. *Comput. Methods Appl. Mech. Eng.* 1999;**180**:203–17. http://dx.doi.org/10.1016/S0045-7825(99)00056-0.

[11] Rozvany GIN. Difficulties in truss topology optimization with stress and local buckling constraints. *Struct. Optim.* 1996;**11**:134–6. http://dx.doi.org/10.1007/BF01376857.

[12] Zhou M. Difficulties in truss topology optimization with stress and local buckling constraints. *Struct. Optim.* 1996;**11**:134–6. http://dx.doi.org/10.1007/BF01376857.

[13] Smith OS. An interactive system for truss topology design. *Adv. Eng. Softw.* 1996;**27**:167–78. http://dx.doi.org/10.1016/0965-9978(96)00010-5.

[14] Achtziger W. Local stability of trusses in the context of topology optimization. Part I: exact modelling. *Struct. Optim.* 1999;**17**:235–46. http://dx.doi.org/10.1007/s001580050056.

[15] Achtziger W. Local stability of trusses in the context of topology optimization. Part II: a numerical approach. *Struct. Optim.* 1999;**17**:247–58. http://dx.doi.org/10.1007/s001580050056.

[16] Stolpe M, Svanberg K. A note on stress-constrained truss topology optimization. *Struct. Multidiscip. Optim.* 2003;**25**:62–4.

[17] Mela K. Resolving issues with member buckling in truss topology optimization using a mixed variable approach. *Struct. Multidiscip. Optim.* 2014;**50**:1037–49. http://dx.doi.org/10.1007/s00158-014-1095-x.

[18] Lim WH, Mat Isa NA. Teaching and peer-learning particle swarm optimization. *Appl. Soft Comput. J.* 2014;**18**:39–58.

[19] Zhang B, Zhang M, Zhang J, Zheng Y. A water wave optimization algorithm with variable population size and comprehensive learning. *Intell. Comput. Theor. Methodol.* 2015;**9225**:124–36.

[20] Patel VK, Savsani VJ. A multi-objective improved teaching – learning based optimization algorithm (MO-ITLBO). *Inf. Sci.* 2014. http://dx.doi.org/10.1016/j.ins.2014.05.049.

[21] Patel VK, Savsani VJ. Optimization of a plate-fin heat exchanger design through an improved multi-objective teaching-learning based optimization (MO-ITLBO). *Chem. Eng. Res. Des.* 2014:1–12. http://dx.doi.org/10.1016/j.cherd.2014.02.005.

[22] Ahrari A, Atai AA, Deb K. Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy. *Eng. Optim.* 2015;**47**:37–41. http://dx.doi.org/10.1080/0305215X.2014.947972.

[23] Ahrari A, Deb K. An improved fully stressed design evolution strategy for layout optimization of truss structures. *Comput. Struct.* 2016;**164**:127–44. http://dx.doi.org/10.1016/j.compstruc.2015.11.009.

[24] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput.-Aided Des.* 2011;**43**(3)303–15.

[25] Rao RV, Savsani VJ, Vakharia DP. Teaching–Learning-Based Optimization: an optimization method for continuous non-linear large scale problems. *Inf. Sci.* 2012;**183**(1)1–15.

[26] Patel VK, Savsani VJ. Heat transfer search (HTS): a novel optimization algorithm. *Inf. Sci.* 2015;**324**:217–46.

[27] Zheng Y-J. Water wave optimization: a new nature-inspired metaheuristic. *Comput. Oper. Res.* 2015;**55**:1–11.

[28] Savsani P, Savsani V. Passing Vehicle Search (PVS): a novel metaheuristic algorithm. *Appl. Math. Model.* 2015. http://dx.doi.org/10.1016/j.apm.2015.10.040.

[29] Ohsaki M. Genetic algorithm for topology optimization of trusses. *Comput. Struct.* 1995;**57**(2)219–25.

[30] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inf.* 1996;**30–4526**:30–45.

[31] Deb K, Gulati S. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elem. Anal. Des.* 2001;**37**:447–65.

[32] De Falco I, Della Cioppa A, Tarantino E. Mutation-based genetic algorithm: performance evaluation. *Appl. Soft Comput.* 2002;**1**:285–99. http://dx.doi.org/10.1016/S1568-4946(02)00021-2.

[33] Baumann B, Kost B. Structure assembling by stochastic topology optimization. *Comput. Struct.* 2005;**83**:2175–84. http://dx.doi.org/10.1016/j.compstruc.2005.02.026.

[34] Kelesoglu O. Fuzzy multiobjective optimization of truss-structures using genetic algorithm. *Adv. Eng. Softw.* 2007;**38**:717–21. http://dx.doi.org/10.1016/j.advengsoft.2007.03.003.

[35] Kunakote T, Bureerat S. Multi-objective topology optimization using evolutionary algorithms. *Eng. Optim.* 2011:37–41.

Please cite this article as: Savsani VJ, et al. Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Designand Engineering* (2016), http://dx.doi.org/10.1016/j.jcde.2016.10.002

[36] Piotrowski AP. Adaptive memetic differential evolution with global and local neighborhood-based mutation operators. *Inf. Sci.* 2013;**241**: 164–94. http://dx.doi.org/10.1016/j.ins.2013.03.060.

[37] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* 2012;**110–111**:151–66. http://dx.doi.org/10.1016/j.compstruc.2012.07.010.

[38] Lu YC, Jan JC, Hung SL, Hung GH. Enhancing particle swarm optimization algorithm using two new strategies for optimizing design of truss structures. *Eng. Optim.* 2013;**45**:1251–71. http://dx.doi.org/10.1080/0305215X.2012.729054.

[39] Nakamura T, Ohsaki M. A natural generator of optimum topology of plane trusses for specified fundamental-frequency. *Comput. Methods Appl. Mech. Eng.* 1992;**94**:113–29. http://dx.doi.org/10.1016/0045-7825(92)90159-h.

[40] Simon D, Omran MGH, Clerc M. Linearized biogeography-based optimization with re-initialization and local search. *Inf. Sci.* 2014;**267**: 140–57. http://dx.doi.org/10.1016/j.ins.2013.12.048.

[41] Wang L, Zou F, Hei X, Yang D, Chen D, Jiang Q. An improved teaching-learning-based optimization with neighborhood search for applications of ANN. *Neurocomputing* 2014;**143**: 231–47. http://dx.doi.org/10.1016/j.neucom.2014.06.003.

[42] Doraghinejad M, Nezamabadi-pour H. Black hole: a new operator for gravitational search algorithm. *Int. J. Comput. Intell. Syst.* 2014: 1–18. http://dx.doi.org/10.1080/18756891.2014.966990.

[43] Das KN, Parouha RP. An ideal tri-population approach for unconstrained optimization and applications. *Appl. Math. Comput.* 2015;**256**: 666–701. http://dx.doi.org/10.1016/j.amc.2015.01.076.

[44] Zhang X, Yuen SY. A directional mutation operator for differential evolution algorithms. *Appl. Soft Comput.* 2015;**30**: 529–48. http://dx.doi.org/10.1016/j.asoc.2015.02.005.

[45] Yi W, Gao L, Li X, Zhou Y. A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems. *Appl. Intell.* 2015;**42**: 642–60. http://dx.doi.org/10.1007/s10489-014-0620-3.

[46] Ali MZ, Awad NH, Suganthan PN. Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization. *Appl. Soft Comput.* 2015;**33**: 304–27. http://dx.doi.org/10.1016/j.asoc.2015.04.019.

[47] Draa A, Bouzoubia S, Boukhalfa I. A sinusoidal differential evolution algorithm for numerical optimisation. *Appl. Soft Comput.* 2015;**27**: 99–126. http://dx.doi.org/10.1016/j.asoc.2014.11.003.

[48] Li X, Yin M. Modified cuckoo search algorithm with self adaptive parameter method. *Inf. Sci.* 2015;**298**:80–97. http://dx.doi.org/10.1016/j.ins.2014.11.042.

[49] Li J-P. Truss topology optimization using an improved species-conserving genetic algorithm. *Eng. Optim.* 2015;**47**:107–28. http://dx.doi.org/10.1080/0305215X.2013.875165.

[50] Yi W, Zhou Y, Gao L, Li X, Mou J. An improved adaptive differential evolution algorithm for continuous optimization. *Expert Syst. Appl.* 2016;**44**:1–12. http://dx.doi.org/10.1016/j.eswa.2015.09.031.

[51] Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. *Adv. Eng. Softw.* 2013;**59**: 19–28. http://dx.doi.org/10.1016/j.advengsoft.2013.03.001.

[52] Dorn WS, Gomory RE, Greenberg HJ. Automatic design of optimal structures. *J. Mec.* 1964;**3**(6)25–52.

[53] Kirsch U. Optimal topologies of truss structures. *Comput. Methods Appl. Mech. Eng.* 1989;**72**:15–28.

[54] Jin P, De-yu W. Topology optimization of truss structure with fundamental frequency and frequency domain dynamic response constraints. *Acta Mech. Solida Sin.* 2006;**19**(3)231–40.

[55] Li L, Liu F. Group search optimization for applications in structural design. *Adapt. Learn. Optim.* 2011;**9**:97–157.

[56] Kaveh A, Ilchi Ghazaan M. Hybridized optimization algorithms for design of trusses with multiple natural frequency constraints. *Adv. Eng. Softw.* 2015;**79**:137–47.

[57] Kaveh A, Mahdavi VR. A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints. *Appl. Soft Comput.* 2015;**34**:260–73.

[58] Bai Y, De Klerk E, Pasechnik D. Exploiting group symmetry in truss topology optimization. *Optim. Eng.* 2009;**10**:331–49. http://dx.doi.org/10.1007/s11081-008-9050-6.

[59] Noilublao N, Bureerat S. Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multi-objective evolutionary algorithms. *Appl. Soft Comput.* 2011;**89**(23–24) 2531–8.

[60] Luh G, Lin C. Optimal design of truss structures using ant algorithm. *Struct. Multidiscip. Optim.* 2008;**36**:365–79.

[61] Savsani P, Jhala RL, Savsani V. Effect of hybridizing Biogeography-Based Optimization (BBO) technique with Artificial Immune Algorithm (AIA) and Ant Colony Optimization (ACO). *Appl. Soft Comput. J.* 2014;**21**:542–53.

[62] Ghosh A, Mallik AK. *Theory of Mechanisms and Machines*. Affiliated East-West Press, New Delhi, India; 1994.

[63] Rao SS. *Engineering Optimization Theory and Practice*, Fourth ed., A John Wiley and Sons Ltd, Hoboken, New Jersey; 2009.

[64] Richardson JN, Bouillard P, Adriaenssens S, Bouillard P, Filomeno R. Multiobjective topology optimization of truss structures with kinematic stability repair. *Struct. Multidiscip. Optim.* 2012;**46**:513–32. http://dx.doi.org/10.1007/s00158-012-0777-5.

[65] Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. *J. Constr. Steel Res.* 2009;**65**:1558–68.

[66] Draa A. On the performances of the flower pollination algorithm – qualitative and quantitative analyses. *Appl. Soft Comput.* 2015;**34**: 349–71.