



# Multi-objective modified heat transfer search for truss optimization

Sumit Kumar<sup>1</sup> · Ghanshyam G. Tejani<sup>2</sup> · Nantiwat Pholdee<sup>3</sup> · Sujin Bureerat<sup>3</sup>

Received: 13 March 2020 / Accepted: 16 March 2020 / Published online: 20 March 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

In this article, a modified version of heat transfer search (HTS) is proposed for multi-objective structural optimization. Contrary to the basic HTS optimizer which activates only one of the three phases of HTS at a time, multi-objective HTS simultaneously exploits the effect of all phases. The proposed modified optimizer is based on the principle of thermodynamics with design solutions being thought of molecules that interact with other molecules of the system itself, and simultaneously with the surrounding molecules through the three modes of heat transfer, namely conduction, convection, and radiation phases. To examine the effectiveness and feasibility of the proposed modification, five truss optimization benchmark problems are used for the performance test. Truss mass minimization and nodal displacement maximization are taken as objectives, while design variables are discrete. The new method along with several recent multi-objective meta-heuristics including ant system, ant colony system, symbiotic organism search, and HTS is used to solve the test problems and compared for the hypervolume and spacing-to-extent indicators. The results reveal that the improved version of HTS is superior to its previous version and the other optimizers. The statistical examination of this study has been performed by conducting Friedman's rank. Results show the dominance of the proposed optimizer performance in comparison with the others.

**Keywords** Structural optimization · Simultaneous search · Meta-heuristics · Discrete design · Engineering problems

## 1 Introduction

Design of optimal structures has been a dynamic field of investigation from a recent couple of decades due to its widespread engineering applications such as bridges, towers, roof supports, building exoskeletons, space frameworks,

mechanical parts, and industries. It has been recognizing that structural optimization with multiple and conflicting design objectives is a challenging task that requires efficient and robust optimization methods to search optimal solutions [29, 56]. The design of a truss basically consists of defining its elemental sizes, shape, and topology so as to get minimum weight or cost under safety constraints [23, 51, 52].

The application of meta-heuristics (MHs) in multi-objective truss design is currently receiving growing attention from researchers around the globe. Owing to the dominance of MHs over classical gradient-based methods as they are easy to study, need low computation cost, and are flexible in nature, recently lots of effort have been made for their implementation in truss optimization problems. Some of the well-recognized MHs for single-objective truss design problems are genetic algorithms (GAs) [12, 17], particle swarm optimization (PSO) [18], simulated annealing [27], teaching–learning-based optimization (TLBO) [51], cuckoo search algorithm [19], tug-of-war optimization algorithm [26], artificial bee colony algorithm [43], sine–cosine optimization algorithm [31], ray optimization [24], and flower pollination algorithm [13]. The literature also demonstrated the enhancement in the performance of original MHs

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

Sumit Kumar  
sumit21sep1990@gmail.com

Nantiwat Pholdee  
nantiwat@kku.ac.th

Sujin Bureerat  
sujbur@kku.ac.th

<sup>1</sup> Department of Mechanical Engineering, GPERI, Gujarat Technological University, Mehsana, Gujarat, India  
<sup>2</sup> Department of Mechanical Engineering, School of Technology, GSFC University, Vadodara, Gujarat, India  
<sup>3</sup> Department of Mechanical Engineering, Faculty of Engineering, Sustainable and Infrastructure Research and Development Center, Khon Kaen University, Khon Kaen 40002, Thailand

with further modification, hybridization, integration, and improvement in solving more intricate truss design problems. Some prominent optimizers are modified symbiotic organisms search (SOS) [23], modified GA [25], modified TLBO [6], improved dolphin echolocation algorithm [20], hybrid gray wolf optimizer and self-adaptive differential evolution [35], improved fireworks algorithm [21], hybridized passing vehicle search [57], improved harmony search algorithms [14], and adaptive SOS [51].

Often, a designer poses a real-world truss design problem with more than one objective to be optimized, which is called multi-objective optimization. Such a problem has multiple-objective functions usually conflicting in nature causing design more intricate. Contrary to single-objective design which has one solution, a multi-objective design problem leads to a set of optimal solutions. This optimal set is termed as a Pareto optimal set if it is viewed in the design domain. Alternatively, it is called Pareto front if it is observed in the objective domain. The literature demonstrates the superiority of MHs in finding these non-dominated optimal solutions set in truss design problems due to their simplicity and flexibility. More importantly, MH can search for a Pareto front within one run. For dealing with multi-objective optimization problem, from the last two decades, a number of multi-objective meta-heuristics (MOMHs) have been developed such as multi-objective GA [17], multi-objective PSO [7], multi-objective immune optimizer [28], multi-objective symbiotic organism search [36], and NSGA-II [15]. The first objective is usually truss weight an indicator for cost, while another objective function can be added for structural performance maximization. The second objective thus can be stress, natural frequency, structure stiffness, and nodal displacement. Owing to more than one optimal solution set, more than one distinct optimization objective and further two distinct search arena multi-objective optimization problems constitute, inefficient MOMHs may fail in obtaining a reliable solution. Consequently, modification of and improvement in MOMHs are always interesting subjects for numerous researchers. Some successful modified and improved optimizers are modified dragonfly optimization optimizer [3], modified adaptive SOS [58], hybrid mutation PSO [62], improved multi-objective GA [34], improved multi-objective PSO [44], hybrid multi-objective evolutionary optimizer [63], multi-objective improved cuckoo search optimizer [30], improved multi-objective JAYA optimization optimizer [39], modified multi-objective Pareto genetic optimizer for [8], modified multi-objective game theory optimizer [40], modified multi-objective immune optimizer [45], hybrid multi-objective modified artificial bee colony, and cuckoo search optimizers [64].

A recently developed state-of-the-art and highly efficient MH called heat transfer search (HTS) [37] based on the principle of the heat transfer process is modified and

implemented for multi-objective structure optimization problems. After HTS initial invention, it has been implemented in various optimization design problems [10, 16, 22, 41]. Furthermore, it has been modified and improved by numerous researchers [32, 33, 46, 53, 58]. The work presented by Savsani et al. [47] displays that MOHTS can accomplish results with more diverse Pareto fronts compared to NSGA-II. Also, the correlation of Pareto front extreme points illustrates the robustness of MOHTS being better than that of multi-objective uniform GA variety, integrated PSO–GA-based multi-objective EA and NSGA-II. In another finding, Tawhid and Savsani [50] proposed an effective  $\epsilon$ -constraint multi-objective HTS algorithm, which results in superiority to other multi-objective forms of GA, PSO, DE, and WCA. Recently, Tejani et al. [56] implemented the HTS algorithm in a multi-objective truss optimization problem subject to discrete design variables. The effectiveness of the algorithm was examined by different benchmark problems, while it is compared with multi-objective ant system (MOAS), multi-objective ant colony system (MOACS), and multi-objective symbiotic organism search (MOSOS). The outcome demonstrates the dominance of HTS over the others. The MOHTS algorithm is applied for the investigation on optimum heat exchanger [42]. The findings show good promise between optimization and analysis of experiments. Shah et al. [48] analyzed a nanoscale irreversible Stirling engine by using HTS with two and three conflicting objectives. Prajapati and Patel [38] implemented HTS for the optimization of the nanofluid-based organic Rankine cycle with multiple objectives of thermal efficiency maximization and leveled energy cost minimization.

Accordingly, there is a gap for further research and analysis as the HTS optimizer has just been proposed. Moreover, as MOHTS has limited investigation records in the literature, its modification should be investigated for various problems. As a result, this examination is proposed to modify the viability of MOHTS by consolidating synchronized transmission of heat by conduction, convection, and radiation of fundamental HTS during the search process to maintain a good balance between local intensification and global diversification.

In this study, the proposed MOMHTS optimizer is applied for multi-objective truss optimization problems with objective functions being weight minimization and maximization of nodal displacement. Five constrained benchmark truss optimization problems (i.e., 10-, 25-, 60-, 72-, and 942-bar trusses) are taken into account for checking the potential of MOMHTS. Moreover, the obtained results of MOMHTS are compared with other MOMHs, which manifest the dominance of the proposed modification.

The rest of the study is structured as follows: Sect. 2 illuminates the basic HTS with all three phases. Section 3 gives details about the proposed MOMHTS optimizer.

Section 4 defines the multi-objective design problem with its formulation. Section 5 comprises a detailed discussion about the truss design problems and result obtained. The suggested work is recapitulated in Sect. 6 followed by its future prospects.

## 2 Heat transfer search (HTS)

HTS is one of the advanced meta-heuristic proposed by Patel and Savsani [37]. HTS is inspired by the basic principle of heat transfer and thermodynamics. The law implies that, with conduction, convection, and radiation modes of heat transfer, a thermal system can achieve the thermal equilibrium within a system or its surroundings to make the system stable. Consequently, all three modes of heat transfer systems are assumed as the search procedures of the HTS optimizer. Here, it is presumed that all three modes will be fairly likely to participate in the search process. The equal selection probability of each mode in the search process is regulated in each iteration by the uniformly distributed random number ‘ $P$ ,’ which ranges from 0 to 1. In the search process, conduction mode is picked when the value is  $P \in [0 - 0.3333]$ , radiation mode is selected when  $P \in [0.3333 - 0.6666]$ , and convection mode is selected if  $P \in [0.6666 - 1]$ . Thus, based on the values of  $P$ , during the search process, the obtained results are updated as per the executed mode of heat transfer in each iteration.

In the HTS algorithm, the population resembles various molecules that experience the heat transfer process, while the design variables show the temperature levels of various molecules. Herein, the energy level of the system molecules is representing the value of the objective function while the surrounding mimicked the best solution. A population size is equivalent to ‘ $n$ ’ number of molecules, and the design variables are equivalent to temperature level ‘ $m$ .’ In each generation ‘ $g$ ’ the population is reproduced by randomly selecting one of the modes. The revised result in HTS will only be approved if it has a superior functional value.

### 2.1 Thermal conduction mode

In conduction, the molecules exchange heat due to the conduction mode between them. For achieving the thermal equilibrium, the molecules with higher energy state transfer energy to the lower energy-level molecules. The system and surrounding molecules can also exchange heat if they are in physical contact. This mode of the HTS algorithm is divided into two segments based on the number of iteration ‘ $i$ .’ The

formulation for the new solution is shown in Eqs. (1) and (2):

Segment 1: when  $i \leq I_{\max}/CDF$

$$S'_{a,\mu} = \begin{cases} S_{b,\mu} + (-P^2 \times S_{b,\mu}), & \text{if } F(S_a) > F(S_b) \\ S_{a,\mu} + (-P^2 \times S_{a,\mu}), & \text{if } F(S_a) < F(S_b) \end{cases} \quad (1)$$

Segment 2: when  $i > I_{\max}/CDF$

$$S'_{a,\mu} = \begin{cases} S_{b,\mu} + (-p_\beta \times S_{b,\mu}), & \text{if } F(S_a) > F(S_b) \\ S_{a,\mu} + (-p_\beta \times S_{a,\mu}), & \text{if } F(S_a) < F(S_b) \end{cases} \quad (2)$$

In the above equations,  $S'_{a,\mu}$  symbolizes the updated molecules;  $a = 1, 2, 3, \dots, n$ ;  $b$  represents a solution which is selected at random;  $a \neq b$ ;  $b \in (1, 2, 3, \dots, n)$ ;  $\mu$  is a design variable index which is selected randomly;  $\mu \in (1, 2, \dots, m)$ ;  $i$  is the current iteration;  $I_{\max}$  is the maximum number of iteration;  $CDF$  is the conduction factor; probability variable is  $P$  where  $P \in [0, 0.3333]$ ;  $p_\beta$  is an arbitrary number varying from 0 to 1;  $P^2$  and  $p_\beta$  represent the Fourier’s equation [11]; conductance parameters  $S_a$  and  $S_b$  signify the temperature change of molecules; and  $CDF$  is assigned as 2 to balance the intensification and diversification [16, 37, 41]. It should be noted that, in each iteration, only one design variable is modified during the conductive process.

### 2.2 Thermal convection mode

In this HTS mode, the system continuously seeks to eliminate the existing energy-level disparity between the system and the surrounding by means of convection heat transfer. The system molecules (at  $S_{\text{mean}}$ ) interrelate to establish thermal equilibrium with the surrounding ( $S_{\text{surr}}$ ). The updated solution is obtained by the mathematical formulation as shown in Eqs. (3) and (4):

$$S'_{a,\mu} = S_{a,\mu} + P \times (S_{\text{surr}} - S_{\text{mean}} \times TCF) \quad (3)$$

$$TCF = \begin{cases} \text{abs}(P - p_\beta), & \text{if } i \leq I_{\max}/COF \\ \text{round}(1 + p_\beta), & \text{if } i > I_{\max}/COF \end{cases} \quad (4)$$

where  $S'_{a,\mu}$  serves as an updated solution;  $a = 1, 2, 3, \dots, n$ ;  $\mu \in (1, 2, \dots, m)$ ;  $i$  is the function evaluations;  $COF$  is a convection factor; probability variable is  $P$  where  $P \in [0.6666, 1]$ ;  $p_\beta$  is an arbitrary number varying from 0 to 1;  $P^2$  and  $p_\beta$  represent the Newton’s law of cooling [11] convection parameters;  $S_{\text{surr}}$  is the surrounding temperature considered as a reference (the best solution) which remains constant;  $S_{\text{mean}}$  denotes the mean temperature of the system which changes during convection;  $TCF$  is a temperature change factor to trade-off between exploitation and

exploration in the convection phase; and COF is assigned 10 [16, 37, 41].

### 2.3 Thermal radiation mode

In this process, due to its temperature level, heat transfer occurs due to radiation released in the form of electromagnetic waves. The system, therefore, interacts with the surrounding temperature or within the system (i.e., another solution) in order to achieve a thermal equilibrium state. This is governed by the Stefan–Boltzmann law of thermodynamics [11]. Here, the solution is updated continuously according to the mathematical formulation given in Eqs. (5) and (6):

$$S'_{a,\mu} = \begin{cases} S_{a,\mu} + P \times (S_{b,\mu} - S_{a,\mu}), & \text{if } F(S_a) > F(S_b) \\ S_{a,\mu} + P \times (S_{a,\mu} - S_{b,\mu}), & \text{if } F(S_a) < F(S_b) \end{cases}; \quad \text{if } i \leq I_{\max}/RDF \quad (5)$$

$$S'_{a,\mu} = \begin{cases} S_{a,\mu} + p_\beta \times (S_{b,\mu} - S_{a,\mu}), & \text{if } F(S_a) > F(S_b) \\ S_{a,\mu} + p_\beta \times (S_{a,\mu} - S_{b,\mu}), & \text{if } F(S_a) < F(S_b) \end{cases}; \quad \text{if } i > I_{\max}/RDF \quad (6)$$

where  $S'_{a,\mu}$  symbolizes the updated molecules;  $a = 1, 2, 3, \dots, n; \mu \in (1, 2, \dots, m); a \neq b; b \in (1, 2, 3, \dots, n)$ ;  $b$  represents a randomly selected molecule;  $i$  is the current iteration; a probability variable is denoted as  $P \in [0.3333, 0.6666]$ ;  $p_\beta$  is an arbitrary number varying from 0 to 1;  $P^2$  and  $p_\beta$  represent the Stefan–Boltzmann equation radiation parameters; and  $S_a$  and  $S_b$  signify the temperature difference in molecules of the system and the surrounding, respectively. The radiation factor is  $RDF$  which is set as 2 [16, 37, 41] to balance the intensification and diversification of the search process in the radiation phase. Here, all the design variables are modified during the course of the iteration process. The procedure of the MHTS optimizer is detailed in Fig. 1.

## 3 Multi-objective modified heat transfer search (MOMHTS)

As a general rule, a MH optimizer can be efficient only if it poses some particular competencies. One of these is the potential to generate new solutions that can usually improve previous or existing solutions and should cover important areas of search where a global optimum can possibly be found. Another competency is that an optimizer should be able to leave the trap of local optima [60]. A good fusion of the above competencies under appropriate conditions will lead to the excellent performance of MHs. This often needs balancing two important mechanisms of as MH: exploration and exploitation (or diversification and intensification) [5, 59, 60]. Empirical knowledge of popular MHs and

simulations of their convergence habit reveals that intensification tends to increase the convergence rate. Exploration, on the other hand, tends to reduce the rate of convergence. Notwithstanding, overly exploration augments the likelihood of discovering global optimality but with lesser efficiency, whereas large exploitation leads to premature convergence. Hence, the correct quantity of exploration and the accurate degree of exploitation should be in a fine balance [61]. This issue itself, however, is an unresolved task of optimization.

In HTS, the system molecules interact with the other system molecules and surrounding molecules through the transfer of heat to reduce the thermal imbalance. Also, at an instant energy transfer process, it is assumed to be done through one of the three modes of HTS. However, accord-

ing to Patel and Savsani [37], the radiation process is more powerful in solving polynomial functions, while, for linear and nonlinear functions, the convection and conduction processes are efficient in providing solutions, respectively. Consequently, throughout the optimization process, if any phase gets more chance to work, then it will work efficiently for only one kind of function but might not work well for the rest. Nevertheless, a system transfers heat concurrently to speed up the thermal balance.

To alleviate these demerits, we come up with a modified version of HTS for multi-objective truss optimization leading to MOMHTS. In this modification, heat transfer proposed through all three modes in the fundamental HTS is synchronized. As the new solution in HTS is very much motivated by the mean solution, randomly selected solution, and the best solution of the population, there is a higher chance of premature convergence and local optima stagnation. This is due to the high proximity of solutions to each other. Therefore, based on the above facts we modified the basic HTS through the inclusion of synchronous heat transfer search, which results in a good balance between local intensification and global diversification of the optimizer. The details about the suggested modification are as follows:

### 3.1 Synchronous heat transfer

In the fundamental HTS optimizer, the energy interaction occurs in the form of heat between system molecules and surrounding molecules for achieving thermal equilibrium. It is assumed that this energy interaction is taken

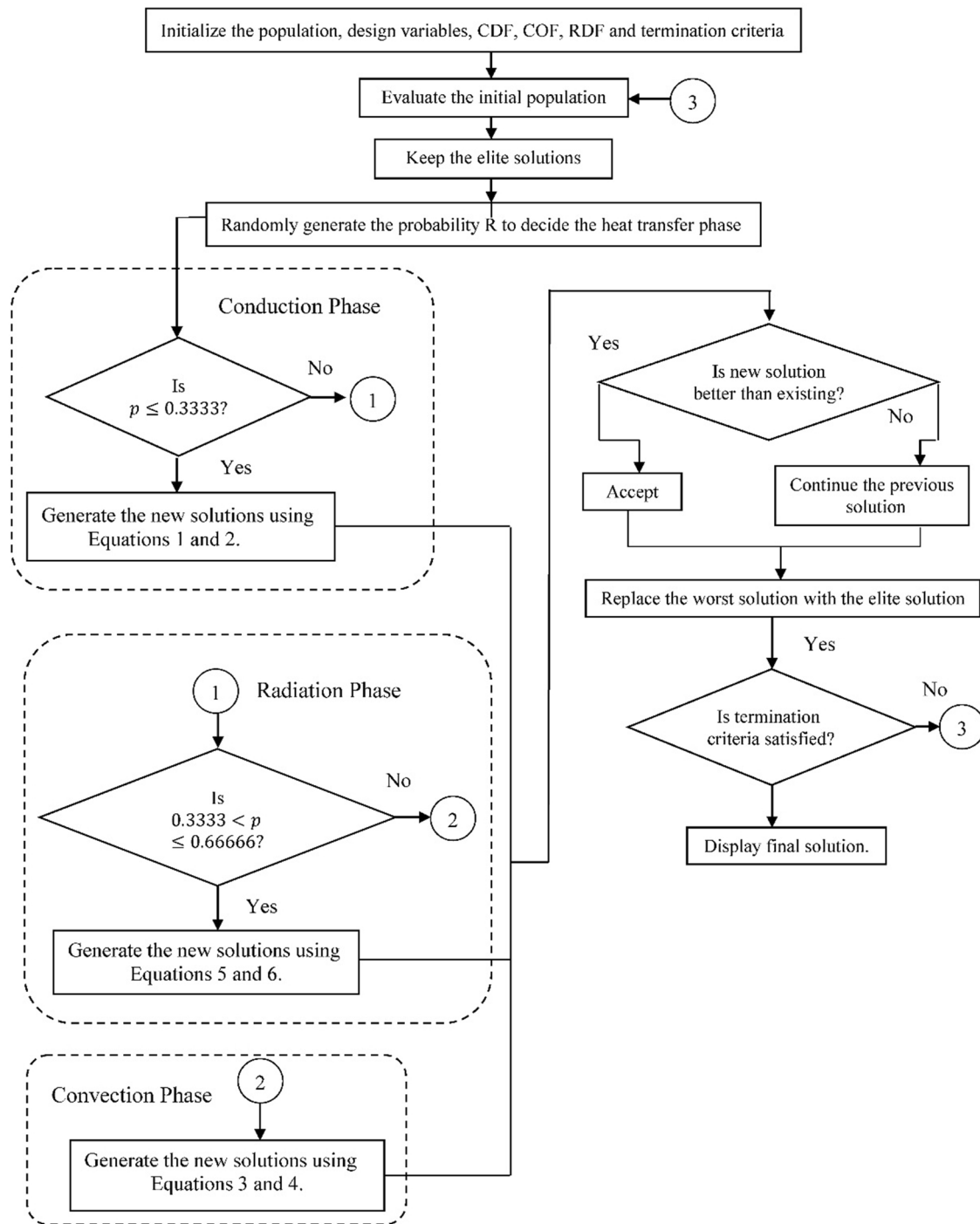


Fig. 1 The heat transfer search algorithm

into account because of one of the three phases with equal likelihood throughout for all generations. However, heat transfer is likely to occur due to one or more than one of the three heat transfer modes being combined. Therefore, in the modified HTS optimizer, synchronized heat transfer is implemented resulting in speeding up the search process. The possibility of heat transfer modes in this

state depends on probability factor for conduction mode (PFCD), probability factor for convection mode (PFCV), and probability factor for radiation mode (PFRD). Here, the values of these factors vary from 0 to 1, i.e.,  $PF_{CD}, PF_{CV}, PF_{RD} \in [0, 1]$  and  $PF_{CD} + PF_{CV} + PF_{RD} = 1$ . Hence, during the activation of the conduction mode of heat transfer, the first one-third of the molecules in a population are

updated. Similarly, the second one-third of solutions are updated during the activation of radiation mode, and the remaining solutions are updated using convection. In the original HTS optimizer, the conduction mode works better for nonlinear functions. The convection mode works more effectively for linear function, whereas the radiation mode works more efficiently for polynomial function. As a consequence, all the three operators should be activated in the synchronized form during optimization to obtain a more efficient HTS algorithm [22, 37]; Tejani et al. [52–54].

With such view, the  $PF_{CD}$ ,  $PF_{CV}$ , and  $PF_{RD}$  values are set at 0.3333 to take into account the effect of equal probability of each mode of energy interaction. Furthermore, to implement all three modes concurrently here three variables (i.e.,  $P_1$ ,  $P_2$ , and  $P_3$ ) are introduced to substitute the probability variable ( $P$ ) where  $P_1$  is a conduction probability variable,  $P_2$  is a radiation probability variable, and  $P_3$  is a convection probability variable. This modification’s mathematical formulation is specified in the algorithm’s implementation steps. The procedure of the MOMHTS optimizer is detailed in Fig. 2.

### 4 Problem definition

Essentially, almost all problems in the physical world are ideally suitable to be shown as multi-objective optimization problems consisting of multiple design objectives with diverse nature. Moreover, these real-world search and optimization problems mostly consist of nonlinear computational problems such as quadratic, cubic and polynomial, which are subject to multiple conflicting objectives. Formerly, due to deficiency of convenient solution techniques to obtain results these problems were transformed into a single objective artificially. The complexity of these problems is due to the generation of multiple optimal solutions, unlike single-objective problems that generate only one solution. Also, the designer has to make a trade-off between a set of multiple optimal solutions Pareto optimal sets as per the obligation. Furthermore, Pareto optimal solutions are typically unknown for a given problem. So, it becomes essential to search for multiple optimal solutions as much as possible within one run, rather than just one solution.

The formulation of the optimization of multi-objective truss problem is shown as follows:

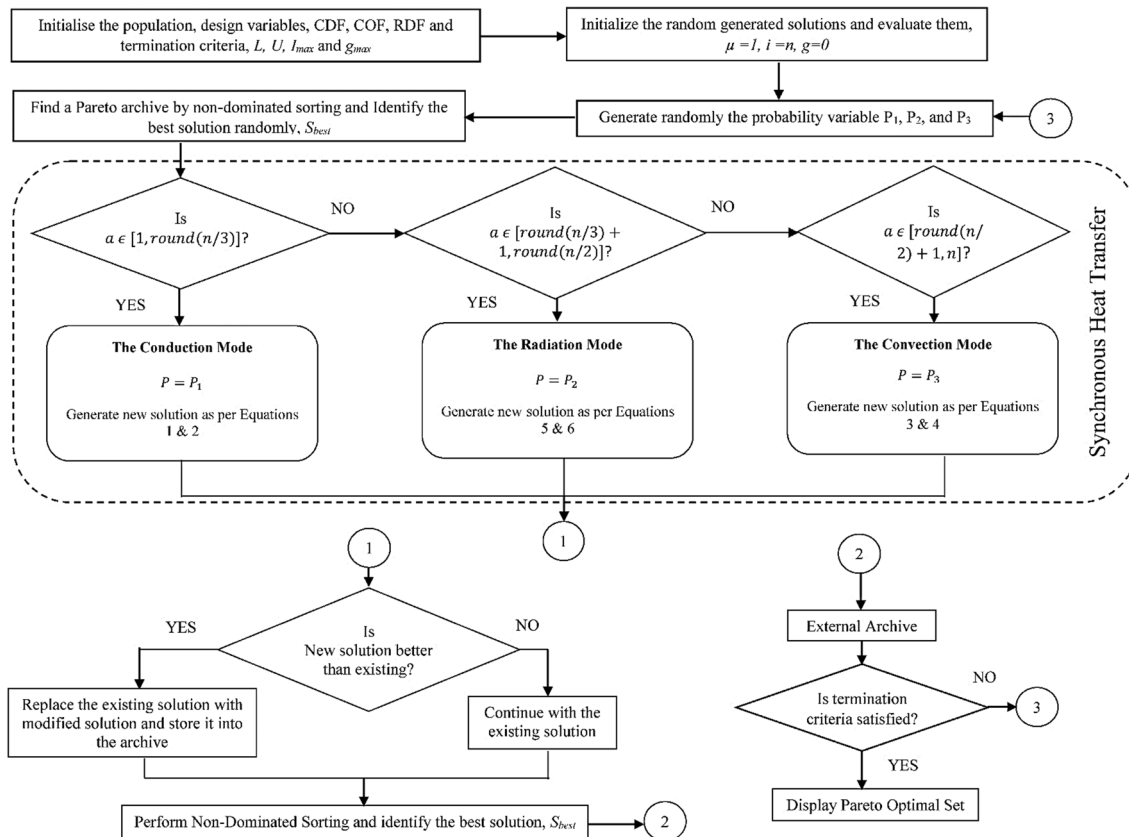


Fig. 2 Modified heat transfer search algorithm

$$\text{Find, } A = \{A_1, A_2, \dots, A_m\} \tag{7}$$

to minimize mass and maximize nodal deflection of truss

$$f_1(A) = \sum_{i=1}^m A_i \rho_i L_i \text{ and } f_2(A) = \max(|\delta_j|)$$

Subject to: Behavior constraints:

$$g(A) : \text{Stress constraints, } |\sigma_i| - \sigma_i^{\max} \leq 0$$

Side constraints:

$$\text{Discrete cross - sectional areas, } A_i^{\min} \leq A_i \leq A_i^{\max}$$

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

Here,  $A_i$  stand for the design variable;  $\rho_i$  represents the mass density;  $L_i$  and  $E_i$  are the elemental length and Young modulus, respectively; and  $\sigma_i$  are elemental stress on the elements. ‘ $\delta_j$ ’ are nodal displacements. The superscripts ‘max’ and ‘min,’ respectively, mean the upper and lower.

### 4.1 Dynamic penalty function

In handling design constraints such as the stress constraints expressed in (7) for use with MHs, a penalty function is the most popular choice as it provides an equivalent unconstrained problem to that with constraints. A good penalty function works in such a way that a feasible solution should have a better penalty function value than an infeasible solution. For two particular feasible solutions, one with a lower objective function is better. For two infeasible solutions, one with less constraint violation is the better. Given that objective function values which are all positive in the design domain, one of the most efficient penalty functions is the multiplication-based penalty function [26], which can be written as:

$$f(X) * (1 + \epsilon_1 * C)^{\epsilon_2}, C = \sum_{i=1}^q C_i, C_i = \left| 1 - \frac{p_i}{p_i^*} \right| \tag{8}$$

where  $p_i$  is the dimension of requirement infringement having the limits as  $p_i^*$ . The number of constraints is presented by  $q$ . The factors  $\epsilon_1$  and  $\epsilon_2$  are pre-specified by a user. In this investigation, the estimations of both  $\epsilon_1$  and  $\epsilon_2$  are set at 3, which were gotten from testing their impact on the parity of the investigation balance [51]; Tejani et al. [52–55].

## 5 Benchmark problems and discussion

The five benchmark truss problems are used to examine the proposed algorithms. These trusses have been examined by numerous researchers, including Angelo et al. [1, 2] and

Tejani et al. [55–57]. This study considers the similar parameters used in the earlier studies [1, 2, 55–57]. Therefore, all five benchmark problems were executed with 100 population size and 50,000 functional evaluations. The considered algorithms are examined for 100 runs. The Pareto front hypervolume (PFHV) test is considered for the assessment. The average of the PFHV is measured to check the convergence rate of the algorithms and the standard deviation (SD) of PFHV is measured to check the algorithms’ reliability.

Likewise, a front spacing ( $S$ ) measure is adopted to compute the comparative separation in the non-dominated set between the consecutive solutions [49]. Since a procedure of Pareto front  $P$  (having  $M$  objective functions) is obtained from using a specific method, the spacing of such a front is computed as:

$$\text{Spacing} = \frac{1}{|P| - 1} \sum_{i=1}^{|P|} (d_i - \bar{d})^2 \tag{9}$$

where  $d_i$  is the Euclidian distance of the vector of objective function ‘ $i$ ’ to its nearest neighbor. The expression  $|P|$  is the number of members in the set  $P$ .  $\bar{d}$  is the mean value of  $d_i$ .

The proportion of front expansion is:

$$\text{Extent} = \sum_{i=1}^M |f_i^{\max} - f_i^{\min}| \tag{10}$$

where  $f_i^{\max}$  and  $f_i^{\min}$  are the minimum and maximum values, respectively, for an  $i$ th objective function shorted from all the members in  $P$ . The lower values of *Spacing* represent the superior Pareto front, while the higher value of *Extent* demonstrates the better Pareto front. The blend of the two indicators prompts another performance metric, which estimates both front spacing and extent, which is characterized as the proportion of spacing to the extent,

$$\text{Front Spacing - to - Extent(FSTE)} = \text{Spacing/Extent.} \tag{11}$$

Equation (11) implies that the superior non-dominated front has a lower *FSTE* value.

Also, a statistical test, Friedman’s rank, is used to rank the various MHs. The truss optimization problems are presented in the following sections.

### 5.1 A 10-bar truss

The first benchmark problem which is broadly used, a 10-bar truss, is illustrated in Fig. 3. The figure also depicts geometrical information (i.e., elements, nodes, loading conditions, constraints, dimensions, etc.) of the truss. The details of design variables, material properties, and constraints are shown in Table 1. The design variables are considered from 42 discrete cross sections (i.e., 1.62, 1.8, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63,

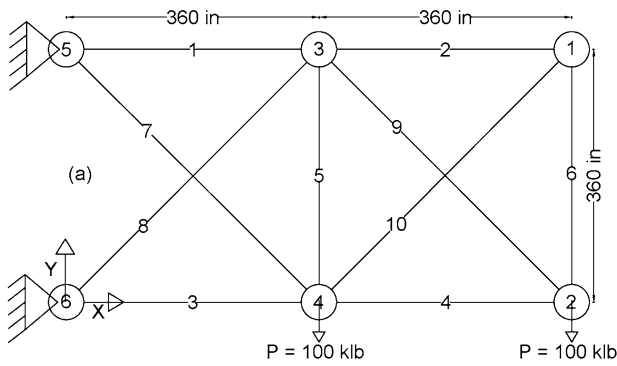


Fig. 3 The 10-bar truss

3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.8, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16, 16.9, 18.8, 19.9, 22, 22.9, 26.5, 30, and 33.5 in.<sup>2</sup>) taken from Angelo et al. [2] and Tejani et al. [55–57].

The statistical results of the 10-bar truss are derived as the PFHV values presented in Table 2. The best PFHV, average PFHV, and SD of PFHVs are measured to check the effectiveness of the presented algorithms. The best

average results found by using AS, ACS, SOS, HTS, and MHTS algorithms are 52,318.34, 55,094.06, 58,292.61, 58,735.88, and 58,849.96, respectively. Also, the SD of PFHVs found by using AS, ACS, SOS, HTS, and MHTS algorithms is 1307.00, 323.84, 149.79, 83.71, and 73.69, respectively. The results reveal that MHTS outperforms the other optimizers, while HTS and SOS rank second and third for the measure of search consistency. Also, a statistical test, Friedman’s rank, is used to compare the results of PFHVs obtained by three algorithms. The Friedman’s ranks for AS, ACS, SOS, HTS, and MHTS algorithms are 100, 200, 300, 413, and 487, respectively. As per the Friedman’s rank test at 95% significance, MHTS beats other algorithms followed by HTS and SOS. The results also show a considerable difference among the considered algorithms.

The front spacing-to-extent (FSTE) metric is tested, and the results are reported in Table 3. The Friedman’s rank shows that MHTS outperforms other MHs followed by HTS and AS, and similar solutions stated as per the mean value of FSTE. Also, MHTS is superior to its original version, HTS.

Table 1 Design considerations of the truss problems

	The 10-bar truss	The 25-bar truss	The 60-bar truss	The 72-bar truss	The 942-bar truss
Design variables	$A_i, i = 1, 2, \dots, 10$	$A_i, i = 1, 2, \dots, 8$	$A_i, i = 1, 2, \dots, 25$	$A_i, i = 1, 2, \dots, 16$	$A_i, i = 1, 2, \dots, 59$
Design constraints	$\sigma^{\max} = 25$ ksi	$\sigma^{\max} = 40$ ksi	$\sigma^{\max} = 40$ ksi	$\sigma^{\max} = 25$ ksi	$\sigma^{\max} = 25$ ksi
Material density	$\rho = 0.1$ lb/in. <sup>3</sup>	$\rho = 0.1$ lb/in. <sup>3</sup>	$\rho = 0.1$ lb/in. <sup>3</sup>	$\rho = 0.1$ lb/in. <sup>3</sup>	$\rho = 0.1$ lb/in. <sup>3</sup>
Modules of elasticity	$E = 10^4$ ksi	$E = 10^4$ ksi	$E = 10^4$ ksi	$E = 10^4$ ksi	$E = 10^4$ ksi

Table 2 The PFHV values for the 10-bar truss

Algorithms	Min.	Max.	Average	SD	Friedman rank	Algorithms’ rank
MOAS	48,719.46	54,558.99	52,318.34	1307.00	100	5
MOACS	54,060.22	55,889.88	55,094.06	323.84	200	4
MOSOS	57,972.16	58,544.08	58,292.61	149.79	300	3
MOHTS	58,447.32	58,859.51	58,735.88	83.71	413	2
MOMHTS	58,631.22	58,953.12	58,849.96	73.69	487	1

Table 3 The FSTE values for the 10-bar truss

Algorithms	Min	Max	Mean	SD	Friedman test	Friedman rank
MOAS	0.005387	0.024711	0.010590	0.003781	355	3
MOACS	0.007219	0.029625	0.014219	0.004558	453	5
MOSOS	0.009190	0.018880	0.010927	0.001145	389	4
MOHTS	0.004924	0.006564	0.005840	0.000407	203	2
MOMHTS	0.003870	0.005331	0.004467	0.000267	100	1



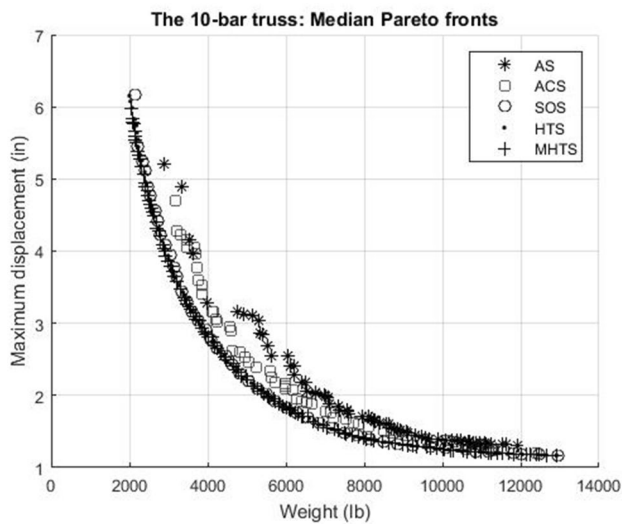


Fig. 4 Median Pareto fronts of the 10-bar truss

The median Pareto fronts of the considered algorithms obtained for the 100 independent runs are displayed in Fig. 4. This can be noted that the median Pareto fronts found for AS and ACS are discontinuous and broken. On the other hand, median Pareto fronts found by using the MHTS, HTS, and SOS are smooth, are continuous, and have a vast array of varied solutions, while the solutions are well spread. Overall, the assessments confirm that MHTS is the better algorithm among the considered algorithms and the considered modification advances the effectiveness of HTS.

### 5.2 A 25-bar 3D truss

The second benchmark problem, a 25-bar 3D truss, is shown in Fig. 5. The details of design variables, material properties, and constraints are depicted in Table 1. Loading conditions are considered as  $P_{x1} = 1 \text{ Klb}$ ,  $P_{y1} = P_{z1} = P_{y2} = P_{z2} = -10 \text{ Klb}$ ,  $P_{x3} = 0.5 \text{ Klb}$ , and  $P_{x6} = 0.6 \text{ Klb}$ . Also, 25 elements of the truss are grouped into eight groups by having structural symmetry with respect to planes  $x-z$  and  $y-z$  [1, 2, 55–57]. The design variables are considered from 30 discrete cross sections (i.e., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3, 3.2, and 3.4 in.<sup>2</sup>) taken from Angelo et al. [1, 2] and Tejani et al. [55–57].

The statistical results of the 25-bar truss are derived as the PFHV values and presented in Table 4. The best PFHV, average PFHV, and SD of PFHVs are measured to check the effectiveness of the presented algorithms. The best average results found by using AS, ACS, SOS, HTS, and MHTS algorithms are 1846.55, 1858.58, 1906.23, 1910.88, and 1913.17, respectively. Also, the SD of PFHVs found by using AS, ACS, SOS, HTS, and MHTS algorithms is 9.51,

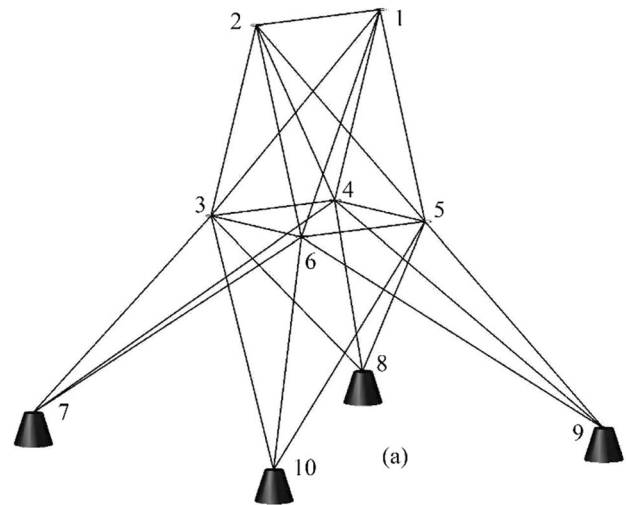


Fig. 5 The 25-bar space truss

13.90, 0.54, 0.51, and 0.38, respectively. The results reveal that MHTS presents the finest convergence and consistency in results' accuracy, while HTS and SOS rank second and third, respectively, for the measure of search consistency. Also, a statistical test, Friedman's rank, is used to compare the results of PFHVs obtained by the considered algorithms. The Friedman's ranks for AS, ACS, SOS, HTS, and MHTS algorithms are 121, 179, 300, 400, and 500, respectively. It is revealed that MHTS is the best, while the second and third best, respectively, is HTS and SOS. The results also show a considerable difference among the considered algorithms.

The comparative FSTE values are given in Table 5. The Friedman's rank MHTS, HTS, and SOS rank first, second, and third, respectively. The mean values of FSTE suggested a similar trend where MHTS outperforms its basic algorithm.

The median Pareto fronts of all the algorithms obtained for the 100 independent runs are shown in Fig. 6. The obtained figure demonstrates the broken and intermittent nature of AS and ACS median Pareto fronts, whereas the MHTS, HTS, and SOS median Pareto fronts are smooth, are continuous, and have a broad range of varied solutions with well spread. Overall, the assessments confirm that the proposed MHTS is the best algorithm among the considered algorithms.

### 5.3 A 60-bar ring truss

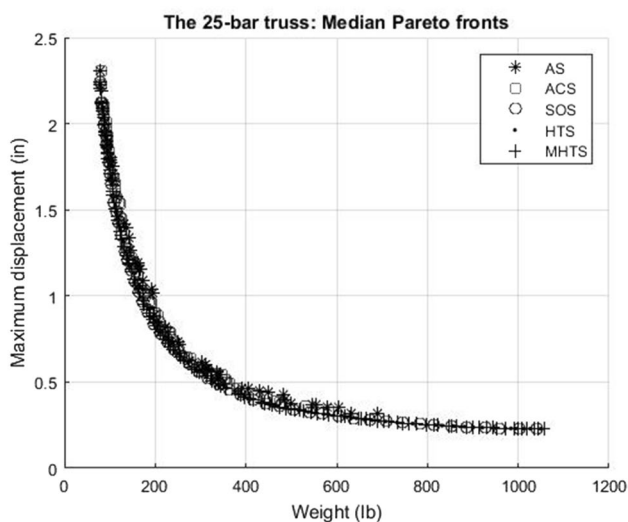
The third benchmark problem, a 60-bar ring truss, is presented in Fig. 7. The details of design variables, material properties, and constraints are given in Table 1. The design variables are considered from 45 discrete cross sections as [0.5, 0.6, 0.7, ..., 4.9] in.<sup>2</sup>. Also, 60 elements of the truss are grouped into 25 groups by considering

**Table 4** The PFHV values for the 25-bar truss

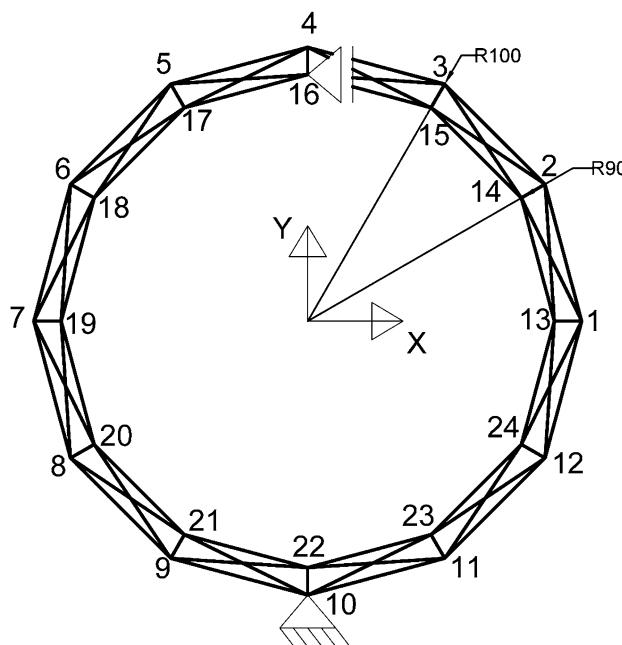
Algorithms	Min.	Max.	Average	SD	Friedman rank	Algorithms' rank
MOAS	1816.72	1869.67	1846.55	9.51	121	5
MOACS	1819.86	1885.73	1858.58	13.90	179	4
MOSOS	1904.56	1907.25	1906.23	0.54	300	3
MOHTS	1909.89	1911.76	1910.88	0.51	400	2
MOMHTS	1911.93	1913.89	1913.17	0.38	500	1

**Table 5** The FSTE values for the 25-bar truss

Algorithms	Min	Max	Mean	SD	Friedman test	Friedman rank
MOAS	0.007937	0.058983	0.022595	0.008424	455	5
MOACS	0.005254	0.044937	0.017026	0.008361	394	4
MOSOS	0.011763	0.013790	0.013255	0.000364	350	3
MOHTS	0.005435	0.007534	0.006498	0.000413	201	2
MOMHTS	0.004375	0.006027	0.005000	0.000283	100	1



**Fig. 6** Median Pareto fronts of the 25-bar truss



**Fig. 7** The 60-bar ring truss

structural symmetry as per Angelo et al. [1, 2] and Tejani et al. [55–57]. Multiple loading conditions are considered as load case 1:  $P_{x1} = -10$  Klb and  $P_{x7} = 9$  Klb, load case 2:  $P_{x15} = P_{x18} = -8$  Klb and  $P_{y15} = P_{y18} = 3$  Klb, and load case 3:  $P_{x22} = -20$  Klb and  $P_{y22} = 10$  Klb.

The comparative results of the 60-bar truss are derived as the PFHV values and presented in Table 6. The best PFHV, average PFHV, and SD of PFHVs are computed to check the effectiveness of the presented algorithms. The best average results found by using AS, ACS, SOS, HTS, and MHTS algorithms are 3179.88, 3106.68, 4293.25, 4316.10, and

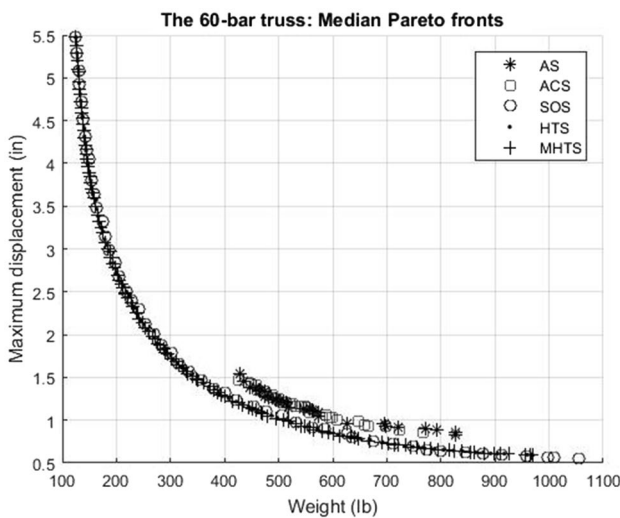
4324.61, respectively. Also, the SD of PFHVs found from the results of AS, ACS, SOS, HTS, and MHTS is 166.65, 74.18, 5.92, 2.04, and 1.51, respectively. The findings indicate that the MHTS obtains the best convergence and performance efficiency, while the second best is HTS. Also, the Friedman’s ranks for AS, ACS, SOS, HTS, and MHTS algorithms are 173, 127, 300, 400, and 500, respectively. MHTS is the best followed by HTS and SOS. Here, the obtained

**Table 6** The PFHV values for the 60-bar truss

Algorithms	Min.	Max.	Average	SD	Friedman rank	Algorithms' rank
MOAS	2465.08	3397.56	3179.88	166.65	173	4
MOACS	2905.27	3276.04	3106.68	74.18	127	5
MOSOS	4271.94	4304.66	4293.25	5.92	300	3
MOHTS	4310.91	4320.11	4316.10	2.04	400	2
MOMHTS	4320.75	4328.66	4324.61	1.51	500	1

**Table 7** The FSTE values for the 60-bar truss

Algorithms	Min	Max	Mean	SD	Friedman test	Friedman rank
MOAS	0.009977	0.133920	0.034915	0.019500	456	5
MOACS	0.007890	0.074504	0.029912	0.013732	438	4
MOSOS	0.010137	0.012876	0.012025	0.000522	306	3
MOHTS	0.005360	0.007235	0.006161	0.000360	200	2
MOMHTS	0.003958	0.005794	0.004766	0.000282	100	1



**Fig. 8** Median Pareto fronts of the 60-bar truss

results from Friedman’s rank test also reflect the appreciable difference between the optimizers being considered.

The comparative FSTE results are shown in Table 7. The Friedman’s rank test shows that MHTS outperforms other MHs followed by HTS and SOS, while similar outcomes are obtained from ranking the mean values of FSTs. Also, MHTS is better than its original version.

Figure 8 displays the median Pareto fronts of the various algorithms obtained for 100 independent runs. This can be noted that the median Pareto fronts found for AS and ACS are irregular and spread in a tiny area. On the other side,

Pareto fronts found by using the MHTS and HTS are continuous, diversified in nature, and well distributed.

### 5.4 A 72-bar space truss

The fourth benchmark problem, a 72-bar 3D truss, is illustrated in Fig. 9. The details of design variables, material properties, and constraints are reported in Table 1. The design variables are considered from 25 discrete cross sections as [0.1, 0.2, 0.3, ..., 2.5] in.<sup>2</sup>. Also, 72 elements of the truss are grouped into 16 by considering structural symmetry as per Angelo et al. [1, 2] and Tejani et al. [55–57]. Multiple loading conditions are considered as load case 1:  $F_{1x} = F_{1y} = 5$  kips and  $F_{1z} = -5$  kips and load case 2:  $F_{1z} = F_{2z} = F_{3z} = F_{4z} = -5$  kips.

The results of the 72-bar truss are compared based on the PFHV values and presented in Table 8. The best PFHV, average PFHV, and SD of PFHVs are measured to check the effectiveness of the presented algorithms. The mean values of PFHV found by using AS, ACS, SOS, HTS, and MHTS algorithms are 2140.24, 2142.38, 2270.93, 2282.66, and 2285.37, respectively. Also, the SD of PFHVs found by using AS, ACS, SOS, HTS, and MHTS algorithms is 10.17, 19.50, 1.81, 0.63, and 0.39, respectively. It shows that MHTS is the best followed by HTS and SOS. The Friedman’s ranks for AS, ACS, SOS, HTS, and MHTS algorithms are 145, 155, 300, 400, and 500, respectively.

The result of comparing FSTE is shown in Table 9. The Friedman’s rank at 95% significant level shows that MHTS, HTS, and SOS are the top three performers in that order, while the mean values of FSTE also give similar results.

Fig. 9 The 72-bar 3D truss

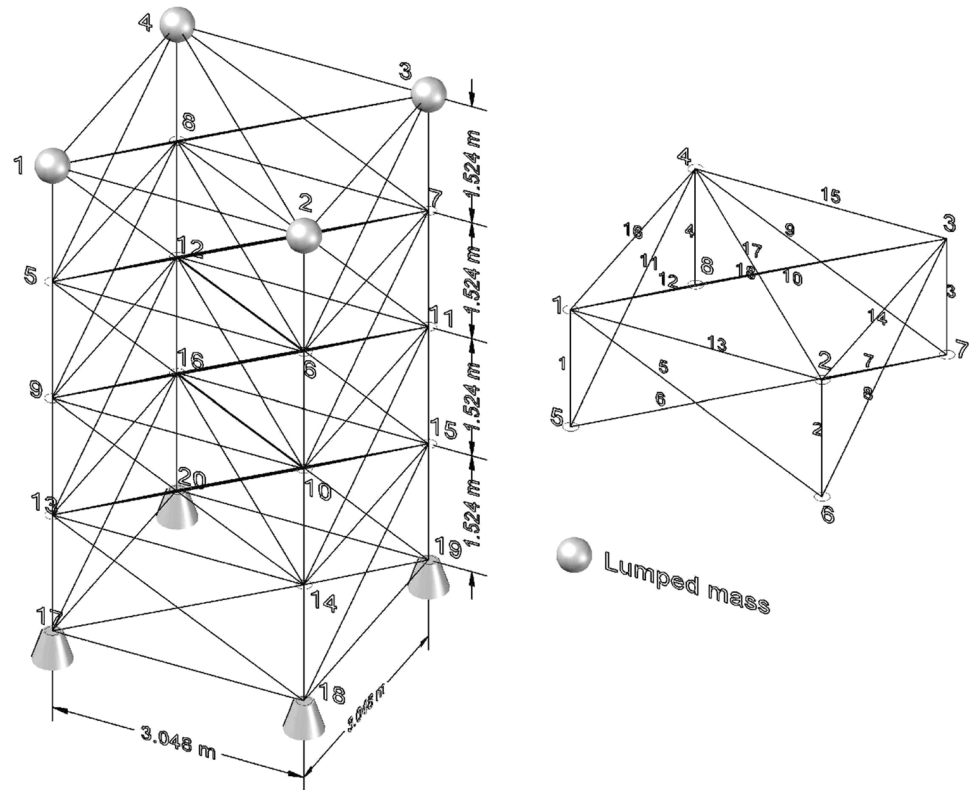


Table 8 The PFHV values for the 72-bar truss

Algorithms	Min.	Max.	Average	SD	Friedman rank	Algorithms' rank
MOAS	2112.10	2162.75	2140.24	10.17	145	5
MOACS	2082.81	2176.18	2142.38	19.50	155	4
MOSOS	2267.43	2274.65	2270.93	1.81	300	3
MOHTS	2280.93	2283.95	2282.66	0.63	400	2
MOMHTS	2284.42	2286.26	2285.37	0.39	500	1

Table 9 The FSTE values for the 72-bar truss

Algorithms	Min	Max	Mean	SD	Friedman test	Friedman rank
MOAS	0.010919	0.043568	0.022728	0.007183	434	4
MOACS	0.007918	0.076088	0.026837	0.013808	441	5
MOSOS	0.013350	0.015643	0.014393	0.000435	324	3
MOHTS	0.005556	0.008281	0.007206	0.000473	201	2
MOMHTS	0.004637	0.007103	0.005425	0.000287	100	1

The median Pareto fronts of all the MHs obtained from performing 100 independent runs are displayed in Fig. 10. This can be noted that the median Pareto fronts found for AS and ACS are discontinuous and broken. On the other hand, Pareto fronts found by using MHTS, HTS, and

SOS are continuously distributed and widely spread. The assessments affirm that MHTS is a superior performer to the other considered algorithms, which implies that the proposed modification improves the HTS performance.

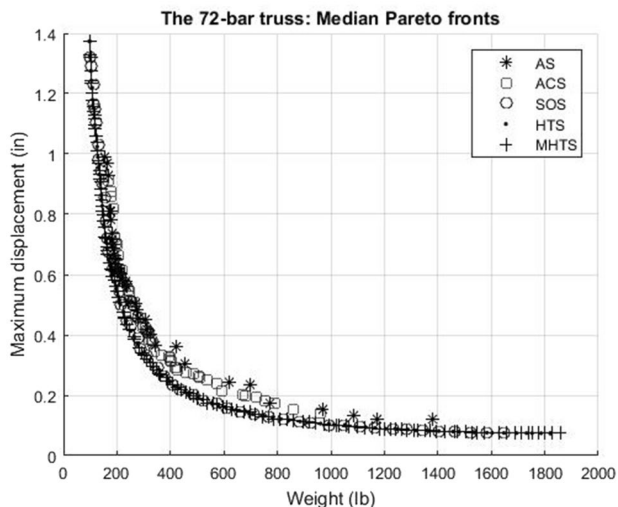


Fig. 10 Median Pareto fronts of the 72-bar truss

### 5.5 A 942-bar tower truss

The fifth benchmark problem, a 942-bar tower truss, is shown in Fig. 11. Vertical loading along the axis of tower center (Z-direction) is  $-3$  kips,  $-6$  kips, and  $-9$  kips at each of the nodes in Sects. 1, 2, and 3, respectively; lateral loading along X-direction is 1.5 kips and 1 kips at each of the nodes on the right-hand side and left-hand side of the tower truss, respectively, and lateral loading along Y-direction is 1 kip at each node, respectively. The details of design variables, material properties, and constraints are depicted in Table 1. The design variables are considered from 200 discrete cross sections as  $[1, 2, 3, \dots, 200]$  in.<sup>2</sup>. Also, 200 elements of the truss are clustered into 59 groups by considering structural symmetry as per Angelo et al. [1, 2] and Tejani et al. [55–57].

The comparative results of the 942-bar truss based on the PFHV values are illustrated in Table 10. The best PFHV, average PFHV, and SD of PFHVs are measured to check the effectiveness of the presented algorithms. The mean values of PFHV found by using AS, ACS, SOS, HTS, and MHTS algorithms are 67,019,485.86, 67,205,495.61, 78,710,767.20, 83,582,579.55, and 84,034,459.15, respectively. Also, the SDs of PFHV found by using AS, ACS, SOS, HTS, and MHTS are 4,596,003.57, 1,247,239.78, 785,836.41, 135,436.80, and 85,499.11, respectively. The results reveal that MHTS performs the best followed by HTS and SOS. Also, a statistical test, Friedman’s rank, is considered to compare the results of PFHVs obtained by the considered algorithms. As per the rank check of the Friedman at a significant level of 95%, it is clear that the MHTS outperforms other optimizers, while HTS and SOS joined the second and third places in the ranking. This comparative results show the superiority of the proposed algorithm over

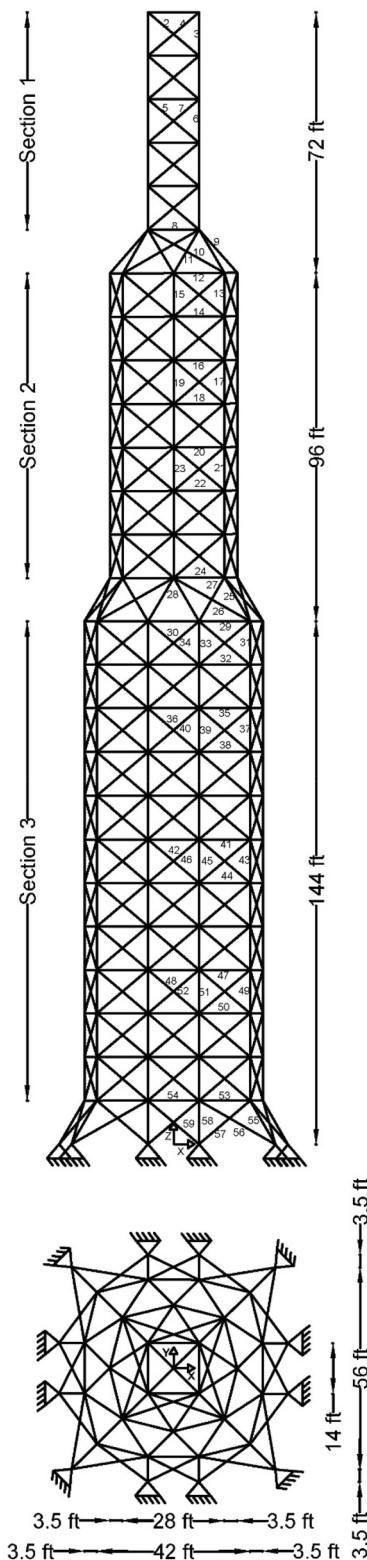


Fig. 11 The 942-bar tower truss

**Table 10** The PFHV values for the 942-bar truss

Algorithms	Min.	Max.	Average	SD	Friedman rank	Algorithms' rank
MOAS	58,098,473.71	74,168,301.18	67,019,485.86	4,596,003.57	153	4
MOACS	63,054,516.72	69,882,829.59	67,205,495.61	1,247,239.78	147	5
MOSOS	76,625,691.80	80,699,769.69	78,710,767.20	785,836.41	300	3
MOHTS	83,267,279.40	83,858,996.35	83,582,579.55	135,436.80	400	2
MOMHTS	83,783,187.69	84,242,890.99	84,034,459.15	85,499.11	500	1

**Table 11** The FSTE values for the 942-bar truss

Algorithms	Min	Max	Mean	SD	Friedman test	Friedman rank
MOAS	0.014273	0.120749	0.042659	0.020981	471	5
MOACS	0.010403	0.079907	0.029013	0.014678	415	4
MOSOS	0.012192	0.016924	0.014856	0.000761	314	3
MOHTS	0.006056	0.008145	0.006795	0.000401	200	2
MOMHTS	0.004479	0.005968	0.005329	0.000300	100	1

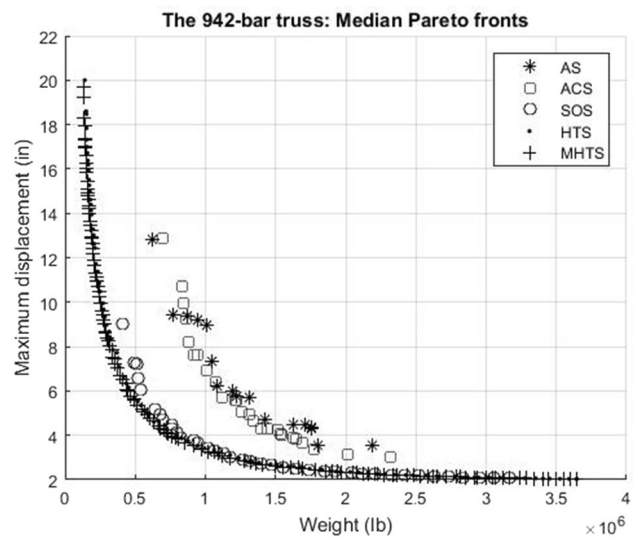
the others for the design problem that can be considered a large-scale problem.

The FSTE values obtained from the various optimizers are given in Table 11. The Friedman’s rank at 95% significant level presents that MHTS, HTS, and SOS are the top three performers in that order. The comparative average results of FSTE also received similar solutions.

Figure 12 illustrates the median Pareto fronts of the optimizers where 100 independent runs for each method are performed. It is understood that in a small territory, the obtained median Pareto fronts are discontinuous and spread out for AS, ACS, and SOS optimizers. Contrary to this, the Pareto fronts are constantly distributed, are stable, and have a variety of different outcomes, when they are discovered using the MHTS and HTS optimizers. This obtained nature supports the supremacy of exploring Pareto fronts using MHTS and HTS over AS, ACS, and SOS. In conclusion, MOMHTS is superior to the methods available in the previous literature. Moreover, from the findings, it can be comprehended that the suggested modification leads to better MOHTS effectiveness.

### 6 Conclusions

This study presented an effective, modified version of a multi-objective HTS optimizer termed MOMHTS for truss optimization. The new algorithm is achieved by means of synchronizing the three heat transfer operators, which is further improved from the previous version of HTS. From the comparative results of the five standard test problems



**Fig. 12** Median Pareto fronts of the 942-bar truss

of truss optimization, it can be concluded that the proposed optimizer is superior to the others found in the literature and its original HTS. The gap between MOMHTS and the rest is even wider when the design problem is large scale. This implies that the proposed method is more suitable to the real world design of truss than the others.

For further work, one can utilize MOMHTS as a potential option for taking care of complex and real-world design problems that cannot be comprehended utilizing the current meta-heuristic optimizers. There is a fairly likely chance of further enhancement in the performance of the

proposed optimizer which can be obtained with further modification, hybridization, integration, and improvement that can be used in solving more intricate truss design problems. Moreover, the application can be extended to the multi-objective problems with conflicting and dynamic constrained design problems.

**Acknowledgements** The authors are grateful for the support from the Thailand Research Fund (RTA6180010).

## References

- Angelo JS, Barbosa HJC, Bernardino HS (2012) Multi-objective ant colony approaches for structural optimization problems. In: Proceedings of the eleventh international conference on computational structures technology. <https://doi.org/10.4203/ccp.99.66>
- Angelo JS, Bernardino HS, Barbosa HJC (2015) Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint. *Adv Eng Softw* 80(C):101–115. <https://doi.org/10.1016/j.advengsoft.2014.09.015>
- Acı Çİ, Gülcan H (2019) A modified dragonfly optimization optimizer for single-and multiobjective problems using Brownian motion. *Comput Intell Neurosci*. <https://doi.org/10.1155/2019/687129>
- Alba E, Dorronsoro B (2005) The exploration/exploitation tradeoff in dynamic cellular genetic optimizers. *IEEE Trans Evol Comput* 9(2):126–142. <https://doi.org/10.1109/TEVC.2005.843751>
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv: CSUR* 35(3):268–308. <https://doi.org/10.1145/937503.937505>
- Camp CV, Farshchin M (2014) Design of space trusses using modified teaching–learning based optimization. *Eng Struct* 62:87–97. <https://doi.org/10.1016/j.engstruct.2014.01.020>
- Coello CC, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on evolutionary computation. CEC'02 (Cat. No. 02TH8600), vol 2. IEEE, pp 1051–1056. <https://doi.org/10.1109/cec.2002.1004388>
- Che ZH, Chiang CJ (2010) A modified Pareto genetic optimizer for multi-objective build-to-order supply chain planning with product assembly. *Adv Eng Softw* 41(7–8):1011–1022. <https://doi.org/10.1016/j.advengsoft.2010.04.001>
- Cuevas E, Echavarría A, Ramírez-Ortegón MA (2014) An optimization optimizer inspired by the States of Matter that improves the balance between exploration and exploitation. *Appl Intell* 40(2):256–272. <https://doi.org/10.1007/s10489-013-0458-0>
- Chaudhari R, Vora JJ, Mani Prabu SS, Palani IA, Patel VK, Parikh DM, de Lacalle LNL (2019) Multi-response optimization of WEDM process parameters for machining of superelastic nitinol shape-memory alloy using a heat-transfer search optimizer. *Materials* 12(8):1277. <https://doi.org/10.3390/ma12081277>
- Cengel YA, Boles MA (2005) *Thermodynamics: an engineering approach*. McGraw-Hill, New York
- Deb K, Gulati S (2001) Design of truss-structures for minimum weight using genetic optimizers. *Finite Elem Anal Des* 37(5):447–465. [https://doi.org/10.1016/S0168-874X\(00\)00057-3](https://doi.org/10.1016/S0168-874X(00)00057-3)
- Draa A (2015) On the performances of the flower pollination optimizer—qualitative and quantitative analyses. *Appl Soft Comput* 34:349–371. <https://doi.org/10.1016/j.asoc.2015.05.015>
- Degertekin SO (2012) Improved harmony search optimizers for sizing optimization of truss structures. *Comput Struct* 92:229–241. <https://doi.org/10.1016/j.compstruc.2011.10.022>
- Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic optimizer: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197. <https://doi.org/10.1109/4235.996017>
- Degertekin SO, Lamberti L, Hayalioglu MS (2017) Heat transfer search optimizer for sizing optimization of truss structures. *Latin Am J Solids Struct* 14(3):373–397. <https://doi.org/10.1590/1679-78253297>
- Fonseca CM, Fleming PJ (1993) Genetic optimizers for multi-objective optimization: formulation discussion and generalization. In: ICGA, vol 93, no. July, pp 416–423. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.9077&rep=rep1&type=pdf>
- Gomes HM (2011) Truss optimization with dynamic constraints using a particle swarm optimizer. *Expert Syst Appl* 38(1):957–968. <https://doi.org/10.1016/j.eswa.2010.07.086>
- Gandomi AH, Talatahari S, Yang XS, Deb S (2013) Design optimization of truss structures using cuckoo search optimizer. *Struct Des Tall Spec Build* 22(17):1330–1349. <https://doi.org/10.1002/tal.1033>
- Gholizadeh S, Poorhoseini H (2016) Seismic layout optimization of steel braced frames by an improved dolphin echolocation optimizer. *Struct Multidiscip Optim* 54(4):1011–1029. <https://doi.org/10.1007/s00158-016-1461-y>
- Gholizadeh S, Milany A (2018) An improved fireworks optimizer for discrete sizing optimization of steel skeletal structures. *Eng Optim* 50(11):1829–1849. <https://doi.org/10.1080/0305215X.2017.1417402>
- Hazra A, Das S, Basu M (2018) Heat transfer search optimizer for non-convex economic dispatch problems. *J Inst Eng (India) Ser B* 99(3):273–280. <https://doi.org/10.1007/s40031-018-0320-1>
- Kumar S, Tejani GG, Mirjalili S (2019) Modified symbiotic organisms search for structural optimization. *Eng Comput* 35(4):1269–1296. <https://doi.org/10.1007/s00366-018-0662-y>
- Kaveh A, Khayatizad 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>
- Kawamura H, Ohmori H, Kito N (2002) Truss topology optimization by a modified genetic optimizer. *Struct Multidiscip Optim* 23(6):467–473. <https://doi.org/10.1007/s00158-002-0208-0>
- Kaveh A, Zolghadr A (2017) Truss shape and size optimization with frequency constraints using tug of war optimization. *Asian J Civ Eng* 7(2):311–333
- Lamberti L (2008) An efficient simulated annealing optimizer for design optimization of truss structures. *Comput Struct* 86(19–20):1936–1953. <https://doi.org/10.1016/j.compstruc.2008.02.004>
- Luh GC, Chueh CH (2004) Multi-objective optimal design of truss structure with immune optimizer. *Comput Struct* 82(11–12):829–844. <https://doi.org/10.1016/j.compstruc.2004.03.003>
- Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscip Optim* 26(6):369–395. <https://doi.org/10.1007/s00158-003-0368-6>
- Meng X, Chang J, Wang X, Wang Y (2019) Multi-objective hydropower station operation using an improved cuckoo search optimizer. *Energy* 168:425–439. <https://doi.org/10.1016/j.energy.2018.11.096>
- Mirjalili S (2016) SCA: a sine-cosine optimizer for solving optimization problems. *Knowl Based Syst* 96:120–133. <https://doi.org/10.1016/j.knsys.2015.12.022>
- Maharana D, Kotecha P (2016a) Simultaneous heat transfer search for computationally expensive numerical optimization. In: 2016 IEEE congress on evolutionary computation (CEC). IEEE, pp 2982–2988. <https://doi.org/10.1109/cec.2016.7744166>

33. Maharana D, Kotecha P (2016b) Simultaneous heat transfer search for single objective real-parameter numerical optimization problem. In: 2016 IEEE region 10 conference (TENCON). IEEE, pp 2138–2141. <https://doi.org/10.1109/tencon.2016.7848404>
34. Narayanan S, Azarm S (1999) On improving multiobjective genetic optimizers for design optimization. *Struct Optim* 18(2–3):146–155. <https://doi.org/10.1007/BF01195989>
35. Panagant N, Bureerat S (2018) Truss topology, shape and sizing optimization by fully stressed design based on hybrid grey wolf optimization and adaptive differential evolution. *Eng Optim* 50(10):1645–1661. <https://doi.org/10.1080/0305215X.2017.1417400>
36. Panda A, Pani S (2016) A symbiotic organisms search optimizer with adaptive penalty function to solve multi-objective constrained optimization problems. *Appl Soft Comput* 46:344–360. <https://doi.org/10.1016/j.asoc.2016.04.030>
37. Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization optimizer. *Inf Sci* 324:217–246. <https://doi.org/10.1016/j.ins.2015.06.044>
38. Prajapati P, Patel V (2019) Multi-objective optimization of CuO based organic Rankine cycle operated using R245ca. In: E3S Web of conferences, vol 116. EDP Sciences, p 00062. <https://doi.org/10.1051/e3sconf/201911600062>
39. Rao RV, Keesari HS, Oclon P, Taler J (2019) Improved multi-objective Jaya optimization optimizer for a solar dish Stirling engine. *J Renew Sustain Energy* 11(2):025903. <https://doi.org/10.1063/1.5083142>
40. Rao SS, Freiheit TI (1991) A modified game theory approach to multiobjective optimization. *ASME J Mech Des* 113(3):286–291. <https://doi.org/10.1115/1.2912781>
41. Raja BD, Patel V, Jhala RL (2017) Thermal design and optimization of fin-and-tube heat exchanger using heat transfer search optimizer. *Therm Sci Eng Progress* 4:45–57. <https://doi.org/10.1016/j.tsep.2017.08.004>
42. Raja BD, Jhala RL, Patel V (2018) Thermal-hydraulic optimization of plate heat exchanger: a multi-objective approach. *Int J Therm Sci* 124:522–535. <https://doi.org/10.1016/j.ijthermalsci.2017.10.035>
43. Sonmez M (2011) Artificial Bee Colony optimizer for optimization of truss structures. *Appl Soft Comput* 11(2):2406–2418. <https://doi.org/10.1016/j.asoc.2010.09.003>
44. Sierra MR, Coello CAC (2005) Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. In: International conference on evolutionary multi-criterion optimization. Springer, Berlin, pp 505–519. [https://doi.org/10.1007/978-3-540-31880-4\\_35](https://doi.org/10.1007/978-3-540-31880-4_35)
45. Sato T, Watanabe K, Igarashi H (2014) A modified immune optimizer with spatial filtering for multiobjective topology optimisation of electromagnetic devices. *Int J Comput Math Electr Electron Eng: COMPEL* 33(3):821–833. <https://doi.org/10.1108/COMPEL-09-2012-0174>
46. Savsani P, Tawhid MA (2018) Discrete heat transfer search for solving travelling salesman problem. *Math Found Comput* 1(3):265–280. <https://doi.org/10.3934/mfc.2018012>
47. Savsani V, Patel V, Gadhi V, Tawhid M (2017) Pareto optimization of a half car passive suspension model using a novel multi-objective heat transfer search optimizer. *Model Simul Eng*. <https://doi.org/10.1155/2017/2034907>
48. Shah P, Saliya P, Raja B, Patel V (2019) A multiobjective thermodynamic optimization of a nanoscale Stirling engine operated with Maxwell-Boltzmann gas. *Heat Transf Asian Res*. <https://doi.org/10.1002/htj.21463>
49. Schott JR (1995) Fault tolerant design using single and multicriteria genetic optimizer optimization (No. AFIT/CI/CIA-95-039). Air force inst of tech Wright–Patterson afb OH. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a296310.pdf>
50. Tawhid MA, Savsani V (2018)  $\epsilon$ -Constraint heat transfer search ( $\epsilon$ -HTS) optimizer for solving multi-objective engineering design problems. *J Comput Des Eng* 5(1):104–119. <https://doi.org/10.1016/j.jcde.2017.06.003>
51. Tejani GG, Savsani VJ, Patel VK (2016) Adaptive symbiotic organisms search (SOS) optimizer for structural design optimization. *J Comput Des Eng* 3(3):226–249. <https://doi.org/10.1016/j.jcde.2016.02.003>
52. Tejani GG, Savsani VJ, Bureerat S, Patel VK (2017) Topology and size optimization of trusses with static and dynamic bounds by modified symbiotic organisms search. *J Comput Civ Eng* 32(2):04017085. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000741](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000741)
53. Tejani G, Savsani V, Patel V (2017) Modified sub-population based heat transfer search optimizer for structural optimization. *Int J Appl Metaheuristic Comput: IJAMC* 8(3):1–23. <https://doi.org/10.4018/IJAMC.2017070101>
54. Tejani GG, Savsani VJ, Patel VK, Mirjalili S (2017) Truss optimization with natural frequency bounds using improved symbiotic organisms search. *Knowl Based Syst*. <https://doi.org/10.1016/j.knsys.2017.12.012>
55. Tejani GG, Pholdee N, Bureerat S, Prayogo D (2018) Multi-objective adaptive symbiotic organisms search for truss optimization problems. *Knowl Based Syst* 161:398–414. <https://doi.org/10.1016/j.knsys.2018.08.005>
56. Tejani GG, Kumar S, Gandomi AH (2019) Multi-objective heat transfer search optimizer for truss optimization. *Eng Comput*. <https://doi.org/10.1007/s00366-019-00846-6>
57. Tejani GG, Savsani VJ, Bureerat S, Patel VK, Savsani P (2019) Topology optimization of truss subjected to static and dynamic constraints by integrating simulated annealing into passing vehicle search optimizers. *Eng Comput* 35(2):499–517. <https://doi.org/10.1007/s00366-018-0612-8>
58. Tejani GG, Pholdee N, Bureerat S, Prayogo D, Gandomi AH (2019) Structural optimization using multi-objective modified adaptive symbiotic organisms search. *Expert Syst Appl* 125:425–441. <https://doi.org/10.1016/j.eswa.2019.01.068>
59. Tejani GG, Savsani VJ, Patel VK, Mirjalili S (2019) An improved heat transfer search optimizer for unconstrained optimization problems. *J Comput Des Eng* 6(1):13–32. <https://doi.org/10.1016/j.jcde.2018.04.003>
60. Yang XS (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley, Hoboken
61. Yang XS, Deb S, Fong S (2014) Metaheuristic optimizers: optimal balance of intensification and diversification. *Appl Math Inf Sci* 8(3):977. <https://doi.org/10.12785/amis/080306>
62. Zhu H, Hu YM, Zhu WD, Fan W, Zhou BW (2020) Multi-objective design optimization of an engine accessory drive system with a robustness analysis. *Appl Math Model* 77:1564–1581. <https://doi.org/10.1016/j.apm.2019.09.016>
63. Zhang W, Wang Y, Yang Y, Gen M (2019) Hybrid multiobjective evolutionary optimizer based on differential evolution for flow shop scheduling problems. *Comput Ind Eng* 130:661–670. <https://doi.org/10.1016/j.cie.2019.03.019>
64. Zhou J, Yao X (2017) A hybrid approach combining modified artificial bee colony and cuckoo search optimizers for multi-objective cloud manufacturing service composition. *Int J Prod Res* 55(16):4765–4784. <https://doi.org/10.1080/00207543.2017.1292064>

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