Latin American Journal of Solids and Structures
www.lajss.org

Heat Transfer Search Algorithm for Sizing Optimization of Truss Structures

Abstract

Heat transfer search (HTS) is a novel metaheuristic optimization algorithm that simulates the laws of thermodynamics and heat transfer. In this study, the HTS algorithm is adapted to truss structure optimization. Sizing optimization searches for the minimum weight of a structure subject to stress and displacement constraints. Three truss structures often taken as benchmarks in the optimization literature are selected here in order to verify the efficiency and robustness of the HTS algorithm. Optimization results indicate that HTS can obtain better designs (i.e. lighter trusses) than most of the state-of-the-art metaheuristic optimizers. The convergence behaviour of HTS also is as good as the other algorithms.

Keywords

Heat transfer search; metaheuristic search algorithms; sizing optimization; truss structures.

S.O. Degertekin [a]
L. Lamberti [b]
M.S. Hayalioglu [a]

[a] Civil Engineering Department, Dicle University, 21280,Diyarbakir, Turkey, sozgur@dicle.edu.tr, hsedat@dicle.edu.tr
[b] Matematica e Management, Politecnico di Bari, 70126, Bari, Italy, luciano.lamberti@poliba.it

http://dx.doi.org/10.1590/1679-78253297

Received 11.08.2016
Accepted 07.12.2016
Available online 09.12.2016

1 INTRODUCTION

The main philosophy of metaheuristic optimization algorithms is to perform a pseudo-random search mimicking some natural phenomenon. Among methods developed in the last two decades, particle swarm optimization (PSO) reproduces the social behaviour of swarms (Kennedy and Eberhart, 1995); harmony search (HS) simulates the natural performance processes of musicians searching for a state of harmony (Geem et. al, 2001); artificial bee colony (ABC) is another swarm intelligence method which mimics the intelligent behaviour of honey bee swarms (Karaboga, 2005); big bang-big crunch (BB-BC) reproduces the process of expansion-contraction of the universe (Erol and Eksin, 2006); charged system search (CSS), developed by Kaveh and Talatahari (2010), utilizes the Newtonian law of mechanics in addition to the electrical physics laws to direct the agents in order to recognize the optimum locations; firefly algorithm (FFA) is inspired by social behaviour of fireflies and the phenomenon of bioluminescent communication (Yang, 2010); teaching-learning-based

optimization (TLBO), developed by Rao et al. (2011), mimics the teaching and learning processes in a classroom, in particular influence of a teacher on learners and the mutual interactions of learners; flower pollination algorithm (FPA) which simulates the pollination process of flowering plants, firstly proposed by Yang (2012); swallow swarm optimization (SSO), developed by Neshat et al. (2013), bases on the analogy between swallow swarm behaviours and optimization problems; water evaporation optimization (WEO), developed by Kaveh and Bakhshpoori (2016), mimics the evaporation of a tiny amount of water molecules adhered on a solid surface with different wettability which can be studied by molecular dynamics simulations.

Various metaheuristic optimization methods have been applied to sizing optimization of truss structures (see, for example, the reviews by Lamberti and Pappalettere (2011), Saka and Dogan (2012), and the textbook by Kaveh (2014)). Just to mention a few examples, Sonmez (2011) proposed an artificial bee colony (ABC) algorithm with an adaptive penalty function approach (ABC-AP) to minimize weight of truss structures; ABC-AP algorithm was found to be competitive in terms of optimized weight but showed very poor convergence capability compared with other metaheuristic algorithms.

Degertekin (2012) developed improved harmony search algorithms called efficient harmony search (EHS) and self-adaptive harmony search (SAHS) for sizing optimization of truss structures. The robustness of the proposed methods was verified by solving four design examples. The results demonstrated that SAHS is very powerful compared to classical harmony search and other metaheuristic optimization methods.

Teaching-learning based optimization (TLBO) was used for optimum design of truss structures by Degertekin and Hayalioglu (2013); the efficiency of the proposed implementation was verified in several truss design examples.

Camp and Farshchin (2014) proposed a modified teaching–learning-based optimization (TLBO) algorithm for optimization of truss structures. Without considering population size, convergence criterion, and penalty function structure, TLBO is parameter insensitive. The performance of above mentioned modified TLBO was found to be equivalent to other metaheuristic methods without applying parameter-based search mechanisms.

Firefly algorithm (FFA) was applied to optimum design of truss structures by Degertekin and Lamberti (2013). FFA proved itself to be very competitive with other metaheuristic optimization methods.

Kaveh et al (2014) hybridized the particle swarm and swallow swarm optimization (HPSSO) to solve mathematical optimization problems and truss weight minimization problems. The results obtained from design examples prove that HPSSO outperforms other PSO variants and is very competitive with state-of-art metaheuristic methods.

Bekdaş et al. (2015) developed an algorithm called flower pollination algorithm (FPA) for sizing optimization of truss structures. The design examples presented in their study showed that FPA could produce better results in some cases. However, it was concluded that detailed parametric study should be performed in order to find the best parameter setting for the FPA algorithm so that it can solve a wider group of structural optimization problems.

Kaveh and Bakhshpoori (2016) tested the water evaporation optimization (WEO) algorithm in six truss design problems from small to normal scale and compared it with the most effective avail-

able state-of-the-art metaheuristic optimization methods. WEO resulted very competitive in terms of solution quality and robustness and the only weak point of the algorithm is its low convergence speed.

A novel metaheuristic search method called heat transfer search (HTS) has been recently developed by Patel and Savsani (2015) for solving optimization problems. HTS simulates the course of action followed by a system to reach thermal equilibrium. The efficiency of HTS was evaluated through 24 mathematical optimization problems with explicit cost function and constraints. A detailed comparison with a variety of metaheuristic methods (besides PSO, ABC and TLBO, also genetic algorithms, differential evolution and biogeography-based optimization were considered) was carried out setting for all algorithms the same limit number of function evaluations or the same convergence tolerance with respect to the target optimum. Numerical results demonstrate the efficiency of HTS compared to other metaheuristic methods.

The main goal of this study is to introduce the HTS algorithm into the optimization of truss structures. The performance of HTS is evaluated by considering three truss structures with 25, 72 and 200 elements. For that purpose, HTS is compared with recently developed metaheuristic optimization methods such as artificial bee colony algorithm with adaptive penalty (ABC-AP), self-adaptive harmony search algorithm (SAHS), teaching-learning based optimization (TLBO), firefly algorithm (FFA), hybrid particle swarm swallow swarm optimization (HPSSO), flower pollination algorithm (FPA) and water evaporation optimization (WEO). The considerable amount of data available in the literature for the selected test problems provides a valuable basis of comparison to evaluate the performance of HTS.

The remainder of this paper is organized as follows: the structural optimization problem is stated in Section 2. The HTS algorithm is explained in Section 3. The implementation of HTS algorithm for optimization of truss structures is presented in Section 4. The design examples are described in Section 5. Finally, some concluding remarks are presented in Section 6.

2 THE STRUCTURAL OPTIMIZATION PROBLEM

The sizing optimization problem of a truss structure including nm members can be formulated as:

$$\text{Find} \quad X=[x_1,x_2,.....,x_{ng}], \qquad x_i^{\min}\le x_i\le x_i^{\max} \qquad i=1,2,....,ng \tag{1}$$

$$\text{to minimize} \quad W(X)=\sum_{i=1}^{ng}x_i\sum_{k=1}^{mk}\rho_k L_k \tag{2}$$

subject to

$$\sigma_m^c\le\sigma_m\le\sigma_m^t, \qquad m=1,2,....,nm \tag{3}$$

$$d_{j,\min}\le d_j\le d_{j,\max}, \qquad j=1,2,.....,ndof \tag{4}$$

where: X is the vector containing the design variables; x_i is the cross-sectional area of the *i-th* group of bars, taken as the *i*-th design variable; $x_i^{\min}$ and $x_i^{\max}$, respectively, are the minimum and maximum values for cross-sectional areas*;* $W(X)$ is the weight of the structure; ng is the number of

design variables, equal to the number of member groups included in the structure; mk is the total-number of members in group k; ϱ_k and L_k, respectively, are the mass density and the length of the k-th member in the i-th group. σ_m is the axial stress of the *m-th* member; σ_m^c and σ_m^t, respectively, are the allowable compression and tension stresses for the *m-th* member. d_j is the nodal displacement of the *j-th* translational degree of freedom, $d_{j,\min}$ and $d_{j,\max}$, respectively, are its lower and upper limits; *ndof* is the number of translational degrees of freedom.

The design constraints given in Eqs. (3-4) are handled by using the following modified feasible-based mechanism, successfully applied to sizing optimization of truss structures (Kaveh and Talatahari, 2009a; Degertekin and Hayalioglu, 2013): (i) Any feasible design is preferred to any infeasible design; (ii) Infeasible designs with a slight constraint violation are taken feasible; (iii) Between two feasible designs, the one having the better objective function value is preferred; (iv) Between two infeasible designs, the one having the smaller constraint violation is preferred.

3 THE HEAT TRANSFER SEARCH ALGORITHM

Heat transfer is the branch of Physics concerned with the exchange of heat between systems having different temperatures. Temperature gradients result in a transport of thermal energy within a system or between systems in thermal contact to each other. Heat transfer occurs because any system attempts to reach the temperature of its surroundings (Hollman, 2010; Çengel, 2008; von Böckh and Wetzel, 2012). Clusters of molecules possess different temperature levels in heat transfer. If a system is thermally unbalanced with itself and/or its neighbouring systems, it attempts to overcome this situation by reaching a state of thermal equilibrium.

Heat transfer consists of three basic mechanisms: conduction, convection and radiation. Conduction is the transfer of energy from the more energetic particles of a substance to the adjacent less energetic ones as a result of interactions between the particles. Convection is a mode of heat transfer between a solid surface and the adjacent liquid or gas that is in motion; it involves the combined effects of conduction and fluid motion. Radiation is the energy emitted by the matter in the form of electromagnetic waves as a result of the changes in the electronic configurations of the atoms or molecules (Hollman, 2010; Çengel, 2008; von Böckh and Wetzel, 2012).

Heat transfer has many application areas such as heating, ventilating and air conditioning systems, thermal power plants, refrigerators and heat pumps, gas separation and liquefaction, cooling of machines, processes requiring cooling or heating, heating up or cooling down of production parts, rectification and distillation plants etc. The detailed information about the thermal equilibrium and heat transfer can be found in the study of Patel and Savsani (2015) and other sources (Hollman, 2010; Çengel, 2008; von Böckh and Wetzel, 2012). Therefore, the same definitions and equations will not be repeated here for the sake of brevity.

The laws of thermodynamics and heat transfer have been incorporated by Patel and Savsani (2015) into the Heat Transfer Search (HTS) metaheuristic optimization algorithm, developed for constrained optimization problems. The HTS algorithm consists of three phases called as 'conduction phase', 'radiation phase' and 'convection phase'. The 'conduction phase', 'radiation phase' and 'convection phase' neutralize the thermal unbalance (i.e. change the energy level) of the system by conduction, radiation and convection heat transfer, respectively. A uniformly distributed random

number (Rn) between 0 and 1 is initially generated in the HTS algorithm in order to decide which phase should be executed. One of these phases is used in an iteration according to the value of Rn and each phase has equal probability between 0 and 1 to be carried out. It is demonstrated that 0–0.3333, 0.3333–0.6666 and 0.6666–1 are suitable intervals for the values of Rn to execute conduction, radiation and convection phases, respectively.

Initial population is generated randomly in the HTS algorithm similar to other population-based optimization algorithms. After that, new designs are produced by using the conduction, convection or radiation phases. If a new trial design yields a better objective function value than the existing one, the previous design is replaced. Otherwise, the original design is left unchanged. Moreover, worst designs of the current iteration are replaced with elite designs of the previous iteration if the elite designs of the previous iteration have better objective function values than the current worst ones. Another rule applied in the HTS algorithm is that if duplicate designs are found in the population after replacing worst designs with elite designs, one of the duplicate designs is modified. For this purpose, a randomly selected design variable of the duplicate design is updated as follows:

$$x_j^{new} = x_j^{old} + r_i x_j^{old} \qquad if \qquad 0.0 < r_i < 0.50 \tag{5}$$

$$x_j^{new} = x_j^{old} - (1 - r_i) x_j^{old} \qquad if \qquad 0.50 \le r_i \le 1.0 \tag{6}$$

where: x_j^{new} and x_j^{old}, respectively, are the new and old values of the selected *j-th* design variable; r_i is a random number generated between 0 and 1.

The analogy between the HTS and optimization of truss structures can be established as follows: different temperatures of molecules represent the different design variables (member groups for a truss design), the energy level of the molecules symbolizes the objective function of the truss structure, the cluster of molecules in the heat transfer represents candidate designs in the population of HTS algorithm. The current best truss design is taken as the surroundings and rest of the truss designs are considered as a system.

3.1 Conduction Phase

The conduction phase is executed if the uniformly distributed random number (Rn) generated is between 0 and 0.3333. In this phase, designs are modified based on the randomly selected design from the population and only one randomly selected design variable is updated. This phase includes two parts. According to the iteration number (it) and the conduction factor (CDF), the first part is executed as follows.

If $it < it_{max}/CDF$ (where it_{max} is the maximum iteration number), designs are updated as follows:

$$X_{j,i}^{new} = X_{k,i}^{old} + CDS_1 \qquad if \qquad W(X_j) > W(X_k) \tag{7}$$

$$X_{k,i}^{new} = X_{j,i}^{old} + CDS_2 \qquad if \qquad W(X_k) > W(X_j) \tag{8}$$

where j=1,2,....,np ; $k \in (1,2,....,np)$ $(j \neq k)$ is a randomly selected truss design from the population, np is the population size (i.e. total number of truss designs in the population), $i \in (1,2,....,ng)$ and i

is the randomly selected design variable, ng is the number of design variables (i.e. number of member groups in a truss design). CDS_1 and CDS_2 are the conduction steps given as follows:

$$CDS_1 = -R_n^2 X_{k,i}^{old} \tag{9}$$

$$CDS_2 = -R_n^2 X_{j,i}^{old} \tag{10}$$

Since the temperature of the system is continuously changing in the heat transfer process, thermal conductivity and conductance also change. The temperature dependent behaviour of conductance is accounted for by the variable Rn which can take any value between 0 and 0.3333 at the beginning of each generation in the conduction phase. Moreover, to exploit the search space, this random variable is modelled by squaring its value so to carry out a fine search (Patel and Savsani, 2015).

If $it \geq it_{max}/CDF$, the second part of the conduction phase is performed as follows:

$$X_{j,i}^{new} = X_{k,i}^{old} - r_i X_{k,i}^{old} \qquad if \qquad W(X_j) > W(X_k) \tag{11}$$

$$X_{k,i}^{new} = X_{j,i}^{old} - r_i X_{j,i}^{old} \qquad if \qquad W(X_k) > W(X_j) \tag{12}$$

where r_i is a randomly generated real number between 0 and 1. As the value of r_i varies between 0 and 1, the optimizer will explore the search space. The conduction factor (CDF) decides the exploration and exploitation tendency of the conduction phase and it is set as 2 (Patel and Savsani, 2015).

Hence, in the conduction phase we randomly select the *k-th* design of population and perturb one design variable of each other *j-th* design included in the population, trying to reach the state of the *k-th* design. Should the *j-th* design be better than the *k-th* design, we perturb the *k-th* design to reach the state of the *j-th* design.

Furthermore, setting CDF=2 results in exploitation and exploration be equally distributed in the conduction phase; Eqs. (9-12) indicate that exploitation characterizes the first half of optimization process. While the latter may seem in contrast with the classical flow of metaheuristic optimization where the best regions of design space found in the exploration phase are locally refined in the exploitation phase, it should be noted that thermal conduction is the most important heath transfer mechanism inside a continuous medium. The candidate designs included in the population converge to the optimum, similar to all temperatures finally reaching the equilibrium temperature. In the initial stages of optimization process, designs cover a larger fraction of design space, similar to having less closely spaced clusters of molecules. Hence, conduction will mainly occur in the neighbourhood of each cluster of molecules, along some preferential direction: this corresponds to have each candidate design locally refined through exploitation by perturbing one variable at a time.

3.2 Convection Phase

In the convection phase, the system tries to reach thermal equilibrium through convection heat transfer. The mean temperature of the system interacts with the surrounding temperature to estab-

lish a thermal balance between the system and the surrounding (the latter denotes the best design in the population). The convection phase is executed for the values of Rn between 0.6666 and 1 (see Section 3). All design variables are simultaneously updated to generate new designs with the following equation:

$$X_{j,i}^{new} = X_{j,i}^{old} + COS_i \qquad if \qquad W(X_j) > W(X_k) \tag{13}$$

where: j=1,2,...,np; i=1,2,...,ng. The COS_i factor is the convection step expressed for the i-th design variable as:

$$COS_i = Rn(x_{s,i} - x_{ms,i} \times TCF) \tag{14}$$

In Eq. (14), Rn is equal to the probability of selecting convection phase. $x_{s,i}$ and $x_{ms,i}$ denote the temperature of the surrounding and the mean temperature of the system, respectively. For truss optimization problems, $x_{s,i}$ is the considered i-th optimization variable of the best design currently included in the population and $x_{ms,i}$ is the mean value for the considered optimization variable averaged over the np agents.

Since the surrounding is treated as a heat sink or heat source in the heat transfer, its temperature remains constant in the current iteration. In order to take into account this effect and attain proper balance between exploration and exploitation, the temperature change factor (TCF) is defined as follows (Patel and Savsani, 2015):

$$TCF = abs(Rn - r_i) \qquad if \qquad it < it_{\max}/COF \tag{15}$$

$$TCF = round(1 + r_i) \qquad if \qquad it \geq it_{\max}/COF \tag{16}$$

where r_i is a random number in the range [0,1]. The value of TCF changes randomly between 0 and 1 in the first part of the convection phase. In the second part of the convection phase, value of TCF changes either as 1 or 2. The COF parameter regulates exploration and exploitation of convection phase. The results of sensitivity analysis carried out by Patel and Savsani (2015) demonstrated that the value of 10 is suitable for the COF parameter. Since population tends to become more and more clustered about the current best record as optimization iterations proceed, $x_{s,i}$ and $x_{ms,i}$ will tend to coincide thus yielding risk of stagnation and premature convergence. In order to avoid this, HTS tries to keep perturbations given to design variables large enough. For that purpose, the difference between $x_{s,i}$ and $x_{ms,i}$ is magnified by increasing TCF in the second part of the optimization process. Therefore, exploration always plays an important role in the convection phase. This is consistent with the physics of the convection phenomenon which affects the whole body surrounded by a heat sink/source. Similar to convection which is driven by the average temperature of the body, the whole population will have to search for a new configuration in the design space.

3.3 Radiation Phase

Radiation phase is executed when the generated random number Rn is between 0.3333 and 0.6666 (see Section 3). In the radiation phase, the system interacts with the surrounding (i.e. best truss design) or within the system itself (i.e. other truss designs) to provide a thermal balance. Similar to

conduction and convection phases, radiation phase consists of two parts. In the first part, where it holds $it \leq it_{max}/RDF$, design is updated as follows:

$$X_{j,i}^{new} = X_{j,i}^{old} + RDS_1 \qquad if \qquad W(X_j) > W(X_k) \tag{17}$$

$$X_{j,i}^{new} = X_{j,i}^{old} - RDS_2 \qquad if \qquad W(X_k) > W(X_j) \tag{18}$$

where: RDF is the radiation factor set equal to 2 as suggested by Patel and Savsani (2015); j=1,2,...,np with $j \neq k$; $k \in (1,2,...,np)$ is a randomly selected design from the population, $i \in (1,2,...,ng)$. All design variables are simultaneously updated in this phase, similarly to what happens in the convection phase.

The radiation steps RDS_1 and RDS_2 are determined as:

$$RDS_1 = Rn(X_{k,i}^{old} - X_{j,i}^{old}) \tag{19}$$

$$RDS_2 = Rn(X_{j,i}^{old} - X_{k,i}^{old}) \tag{20}$$

where Rn is the probability of selecting the radiation phase.

In the second part of radiation phase, where it holds $it \geq it_{max}/RDF$, design is updated as follows:

$$X_{j,i}^{new} = X_{j,i}^{old} + RDS_3 \qquad if \qquad W(X_j) > W(X_k) \tag{21}$$

$$X_{j,i}^{new} = X_{j,i}^{old} - RDS_4 \qquad if \qquad W(X_k) > W(X_j) \tag{22}$$

The values of RDS_3 and RDS_4 steps are now computed as:

$$RDS_3 = r_i(X_{k,i}^{old} - X_{j,i}^{old}) \tag{23}$$

$$RDS_4 = r_i(X_{j,i}^{old} - X_{k,i}^{old}) \tag{24}$$

where r_i is a random number generated in the range [0, 1] for the i-th design variable.

Basically, in the radiation phase, all designs "j" are compared with a randomly selected design "k" and move towards this design if such a movement yields improvement in cost function. The distance between design "j" and "k" is scaled by the same random number Rn between 0.3333 and 0.6666 in the first half of the optimization process and other random numbers r_i, that change for each design variable, in the remaining iterations. This suggests that exploration will characterize the first half of the optimization process (somehow similar to having a constant velocity in particle swarm optimization) while exploitation is carried out later by locally refining the size of the movement for each agent and/or design variable. It should be noted that since random numbers Rn and r_i all belong to the (0,1) uniform distribution, they may have the same probability to be generated and could even coincide. However, the distance $\| \boldsymbol{X}_j^{old} - \boldsymbol{X}_k^{old} \|$ between designs "j" and "k" decreases as we approach the optimum design.

The population updating process is consistent with the physics of the radiation phenomenon: each cluster of molecules can exchange heat with any other cluster of molecules regardless they are in physical contact or not. Radiation is a strongly "individual" process where each part of the system emits or receives thermal waves travelling in the space. It is interesting to compare Eqs. (17-24) that govern the radiation phase with Eqs. (7-12) that govern the conduction phase. Since heath flux intensity increases with thermal gradient, we may think to reach more quickly thermal equilibrium by selecting, for example, regions with very large temperature differences. Let us assume that region "A" is at higher temperature than region "B". If A and B are very far, heat flux originating from A will have to reach intermediate positions C, D etc in the path from A to B before arriving to B. In optimization terns, HTS will have to generate trial designs in the neighbourhood of A to approach the position of B in the design space. These trial designs are reached by the thermal front and in turn will transfer it to other trial designs until reaching position B. As is clear, this an exploitation type mechanism as we cannot perturb too much the current design in order to remain in its neighbourhood. Conversely, the radiation front emitted from A will directly reach B without passing through any intermediate position, i.e. other designs of the population: this appears to be a typical exploration search mechanism.

4 IMPLEMENTATION OF THE HTS ALGORITHM IN TRUSS SIZING OPTIMIZATION

The search mechanism of the HTS algorithm was explained in the previous section. The algorithm includes three phases each of which is divided in two parts whose activation is based on the current number of iterations and depends on the conduction, convection and radiation factors. The number of structural analyses performed in each iteration is equal to the population size *np*. An iteration is completed when the randomly selected phase is performed. The HTS algorithm repeats the search process until the predetermined total number of iterations is performed.

The HTS algorithm for sizing optimization of truss structures developed in this study includes the following steps:

Step 1: Initialize the HTS algorithm parameters: population size (*np*), elite design set size (*ies*), conduction factor (*CDF*), convection factor (*COF),* radiation factor (*RDF*). Set the iteration counter: *it*=0.

Step 2: Randomly generate an initial population consisting of truss designs. Each design variable in a truss design is generated between its minimum ($x^j_{\min}$) and maximum ($x^j_{i\max}$) boundary values using the following equation:

$$x_i^j = x_{i\min}^j + rand(0,1)(x_{i\max}^j - x_{i\max}^j) \qquad i=1,2,...,ng; \qquad j=1,2,...,np \tag{25}$$

Truss designs included in the population are organized into a matrix where each row represents a candidate design. An extra-column with the values of cost function computed for each design may be added to the matrix.

$$x_i^j = \begin{bmatrix} x_1^1 & x_2^1 & . & . & x_{ng}^1 \\ x_1^2 & x_2^2 & . & . & x_{ng}^2 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_1^{np} & x_2^{np} & . & . & x_{ng}^{np} \end{bmatrix} \begin{matrix} \rightarrow W(X^1) \\ \rightarrow W(X^2) \\ \\ \\ \rightarrow W(X^{np}) \end{matrix} \qquad i=1,2,...,ng; \quad j=1,2,...,np \tag{26}$$

Step 3: Calculate the objective function $W(X)$, analyze the truss designs and obtain nodal displacements and member stresses using Eqs. (2-4).

Step 4: Increase the iteration counter, $it=it+1$. Generate a uniformly distributed random number Rn between 0 and 1 in order to decide which heat transfer phase should be performed.

Step 5: If $0 \leq Rn \leq 0.3333$, perform the conduction phase by using Eqs. (7-12).

Step 6: If $0.3333 < Rn < 0.6666$, perform the radiation phase by using Eqs (17-24).

Step 7: If $0.6666 \leq Rn \leq 1.0$, perform convection phase by using Eqs. (13-16).

Step 8: Obtain a new truss design, calculate cost function value $W(X)$ and constraints with Eqs. (2-4). If the new truss design has better (i.e. lighter) $W(X)$ value than the previous one, exchange them. Otherwise, leave the original design unchanged. Use the four constraint handling rules described in Section 2. Repeat this process until all designs in the population are updated.

Step 9: Replace the worst designs of the current iteration with the elite designs of previous iteration if these have better values of cost function than the current worst designs.

Step 10: If duplicate designs exist in the population after replacing worst designs with elite designs, modify one of the duplicate designs with Eqs. (5-6).

Step 11: Stop the search process if the termination criterion is satisfied. Assign the minimum weight truss design with no constraint violation as the final optimum design. Otherwise, go to step 4.

5 TEST PROBLEMS AND OPTIMIZATION RESULTS

The efficiency of the HTS algorithm is tested on three classical truss structures (including, respectively, 25, 72 and 200 elements) commonly used as benchmark in the optimization literature. The results obtained by HTS are compared with those of artificial bee colony with adaptive penalty function (ABC-AP), hybrid big bang-big crunch (HBB-BC), self-adaptive harmony search algorithm (SAHS), teaching-learning based optimization (TLBO), firefly algorithm (FFA), multi-stage particle swarm optimization (MSPSO), hybrid particle swarm and swallow swarm optimization (HPSSO), corrected multilevel and multipoint simulated annealing (CMLPSA), modified teaching learning based optimization (TLBO), flower pollination algorithm (FPA) and water evaporation optimization (WEO).

Thirty independent optimization runs are executed for each test problem starting from thirty randomly generated initial populations. The best design obtained in the thirty optimization runs, the number of structural analyses required by the best run, the average optimized weight and the standard deviation of optimized weight are shown in the tables. The HTS algorithm is coded in

Fortran and optimizations are carried out on a standard PC equipped with a single 2.6 GHz Intel® Pentium Core i5-3320M CPU.

It was demonstrated in the study of Patel and Savsani (2015) that the most suitable values for control parameters of HTS are 2 for *ies*, *CDF* and *RDF*, and 10 for *COF,* respectively. Here, a new sensitivity analysis was performed starting from the same values of control parameters suggested by Patel and Savsani (2015) for mathematical optimization problems. The inherent robustness of HTS algorithm made it possible to immediately obtain very competitive results also for truss sizing problems. Such a behavior has indeed a theoretical explanation. In HTS, conduction, convection and radiation phases are activated depending on the value extracted in the interval (0,1) for the uniformly distributed random number *Rn*. Hence, the three mechanisms practically occur the same number of times in the optimization process. Conduction and radiation phases are anti-symmetric in the sense that the former turns from exploitation to exploration while the latter turns from exploration to exploitation as iterations progress. Since the ideal condition is to reach an exact balance of these two mechanisms, setting CDF=RDF=2 is practically a straightforward option as the iteration history is divided in two equal parts. As far as it concerns the setting of convection factor COF, we found that setting this parameter between 5 and 15 results in small variations of convergence behavior. This can be explained with the following informal argument. Conduction/radiation mechanisms are twice more likely to be selected than convection because the *Rn* number is uniformly distributed. Since convection passes from exploitation to exploration and conduction/radiation mechanisms are anti-symmetric, it is preferable to "limit" the exploitative part of convection phase to have enough search freedom over the whole optimization process. In order to satisfy this goal, we started with setting COF=4, that is equal to the sum (CDF+RDF), and then increased COF to the square power $(CDF+RDF)^2$=16. Interestingly, the average of these bounds is just 10, i.e. the value suggested in literature for the COF parameter.

Population size (*np*) was set equal to 50 after many numerical trials because using smaller populations resulted in a premature convergence of optimization process while larger values than 50 did not improve optimum design significantly. Evidence gathered from sensitivity analysis led to set the maximum number of iterations as 400 for the first and the second design examples and 500 for the last design example, respectively. The maximum number of structural analyses performed by HTS is equal to the product between population size and limit number of iterations: 20000 for the first and second design examples and 25000 for the last design example, respectively.

Although sizing design of truss structures may appear computationally trivial, and multi-start gradient-based optimization can efficiently solve this problem, truss sizing optimization still is a legitimate testing grounds for new global optimization algorithms for at least two reasons. First, gradient-based optimizers may exhibit premature convergence if search starts very far from optimum or from an infeasible region. Interestingly, particle swarm optimization is the metaheuristic algorithm that most suffers from premature convergence because its formulation somehow reflects the line search process of gradient-based optimization (see, for example, Perez and Behdinan (2007)).

Second, there are many references on truss problems in the technical literature. This makes the problem of developing/testing new algorithms more challenging even for simpler problems because any algorithm never applied before to a specific type of optimization problems always should im-

prove existing results or at least reach a very good compromise between capability to find global optimum, fastness and robustness. However, it appears from the literature that practically any state-of-the-art metaheuristic algorithm neither converges to slightly heavier designs than those quoted in the leading studies even for small/average scale problems, nor it can maintain the same level of efficiency in all test cases. In our specific case, the apparent easiness of truss structures is counterbalanced by the existence of local minima besides the global optimum: this happens, for example, of the 25-bar and 200-bar truss problems.

5.1 Spatial 25-Bar Truss Structure

The spatial 25-bar truss structure shown in Fig. 1 is chosen as first design example. The structure is made of aluminium: Young's modulus is 10 Msi while mass density is 0.1 lb/in^3. The structure is subject to the two independent loading conditions listed in Table 1. Because of structural symmetry, bars can be divided in eight groups: the corresponding sizing variables and the allowable stress values for all groups are listed in Table 2. Cross-sectional area of elements must be greater than 0.01 in^2. The displacement of each free node in all coordinate directions must be less than $\pm$0.35 in.

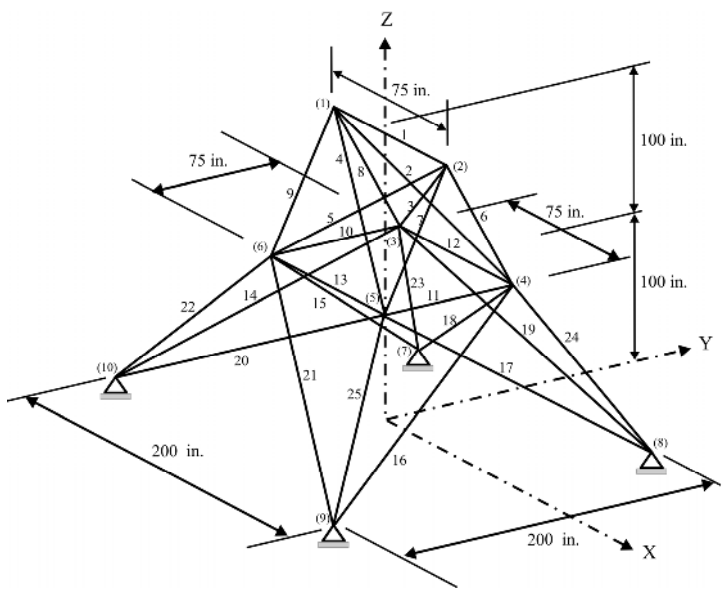

Figure 1: Schematic of the spatial 25-bar truss structure.

	Condition 1			Condition 2		
Node	F_x	F_y	F_z	F_x	F_y	F_z
1	0.0	20.0	-5.0	1.0	10.0	-5.0
2	0.0	-20.0	-5.0	0.	10.0	-5.0
3	0.0	0.0	0.0	0.5	0.	0.
6	0.0	0.0	0.0	0.5	0.	0.

Note: loads are in kips

Table 1: Loading conditions for the spatial 25-bar truss.

Design variables A_i (in^2)	Allowable compressive stress (ksi)	Allowable tensile stress (ksi)
A_1	35.092	40.0
A_2-A_5	11.590	40.0
A_6-A_9	17.305	40.0
A_{10}-A_{11}	35.092	40.0
A_{12}-A_{13}	35.092	40.0
A_{14}-A_{17}	6.7590	40.0
A_{18}-A_{21}	6.9590	40.0
A_{22}-A_{25}	11.082	40.0

Table 2: Allowable stresses in element groups for the spatial 25-bar truss.

The results obtained by the HTS algorithm and other metaheuristic optimizers are compared in Table 3. HTS found an optimum design weighing 545.13 lb after 7653 structural analyses, hence much before the limit number of 20000 analyses set for this test problem. This design is better than those found by ABC-AP (Sonmez, 2011), FFA (Degertekin and Lamberti, 2013), MSPSO (Talatahari et. al, 2013), HPSSO (Kaveh et. al, 2014), TLBO (Camp and Farshchin, 2014), FPA (Bekdaş et. al, 2015) and WEO (Kaveh and Bakhshpoori, 2016), but it is between 0.01 and 0.04 lb heavier than the optimum designs of SAHS (Degertekin, 2012) and TLBO (Degertekin and Hayalioglu, 2013) that are the best quoted in the literature. Furthermore, the number of structural analyses required by HTS for completing the optimization process is the smallest among the metaheuristic algorithms compared in this study.

It should be noted that the designs optimized by HPSSO (Kaveh et. al, 2014) and FPA (Bekdaş et. al, 2015) violate slightly design constraints while HTS converged to a strictly feasible design. Another optimized design often cited in literature is that obtained by Kaveh and Talatahari (2009b): their hybrid big bang-big crunch algorithm found an optimum weight of 545.16 lb within 12500 structural analyses. Since this design violates constraints by 2.06%, it was not included in Table 3 for the sake of brevity.

It can be seen from Table 3 that HTS is robust enough compared with the other metaheuristic algorithms. WEO (Kaveh and Bakhshpoori, 2016), TLBO (Degertekin and Hayalioglu, 2013), FFA (Degertekin and Lamberti, 2013), SAHS (Degertekin, 2012) and HPSSO (Kaveh et. al, 2014) achieved a smaller standard deviation than HTS (between 0.083 and 0.432 lb vs. 0.476 lb) but at a considerably higher computational cost (between 12199 and 25014 vs. only 7653 structural analyses) or even violating optimization constraints (HPSSO and FPA).

The optimization histories plotted in Fig. 2 for HTS and other representative algorithms cover only the first 8000 structural analyses, thus including the 7653 analyses required by HTS to find its best design. It can be seen that the present algorithm could rapidly approach optimum weight (in particular, it required about 6000 structural analyses to find an intermediate design only 0.01% heavier than the final weight of 545.13 lb reported in Table 3). Furthermore, it is very competitive with the other metaheuristic optimizers in terms of convergence speed although the best agent included in the HTS initial population very often yield a larger structural weight than for the other algorithms. WEO (Kaveh and Bakhshpoori, 2016) showed the slowest convergence rate overall

while TLBO (Degertekin and Hayalioglu, 2013) was initially very fast but its convergence rate decreased significantly between 1200 and 2500 structural analyses.

Design variables A_i (in^2)	ABC-AP (Sonmez, 2011)	SAHS Degertekin, 2012)	TLBO (Degertekin and Hayalioglu, 2013)	FFA (Degertekin and Lamberti, 2013)	MSPSO (Talatahari et. al, 2013)	HPSSO (Kaveh et. al 2014)	TLBO (Camp and Farshchin, 2014)	FPA (Bekdaş et al, 2015)	WEO (Kaveh et. al 2016)	HTS This study
A_1	0.011	0.010	0.010	0.010000	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
A_2-A_5	1.979	2.074	2.0712	1.962884	1.9848	1.9907	1.9878	1.8308	1.9184	2.0702
A_6-A_9	3.003	2.961	2.9570	3.030559	2.9956	2.9881	2.9914	3.1834	3.0023	2.970031
A_{10}-A_{11}	0.010	0.010	0.0100	0.010000	0.0100	0.0100	0.0102	0.0100	0.0100	0.01000
A_{12}-A_{13}	0.010	0.010	0.0100	0.010000	0.0100	0.0100	0.0100	0.0100	0.0100	0.01000
A_{14}-A_{17}	0.690	0.691	0.6891	0.6837895	0.6852	0.6824	0.6828	0.7017	0.6827	0.67079
A_{18}-A_{21}	1.679	1.617	1.6209	1.680585	1.6778	1.6764	1.6775	1.7266	1.6778	1.61712
A_{22}-A_{25}	2.652	2.674	2.6768	2.651661	2.6599	2.6656	2.6640	2.5713	2.6612	2.6981
Weight (lb)	545.193	545.12	545.09	545.18	545.16	545.164	545.175	545.159	545.166	545.13
Average weight (lb)	N/A	545.94	545.41	545.394	546.03	545.556	545.483	545.730	545.226	545.47
Std dev (lb)	N/A	0.91	0.42	0.37	0.8	0.432	0.306	0.59	0.083	0.476
Constraint tolerance (%)	None	None	None	None	None	0.0013	None	0.138	None	None
No. struct. analyses	300,000	9051	15318	25014	10800	13326	12199	8149	19750	7653

Table 3: Optimization results for the 25-bar truss problem.

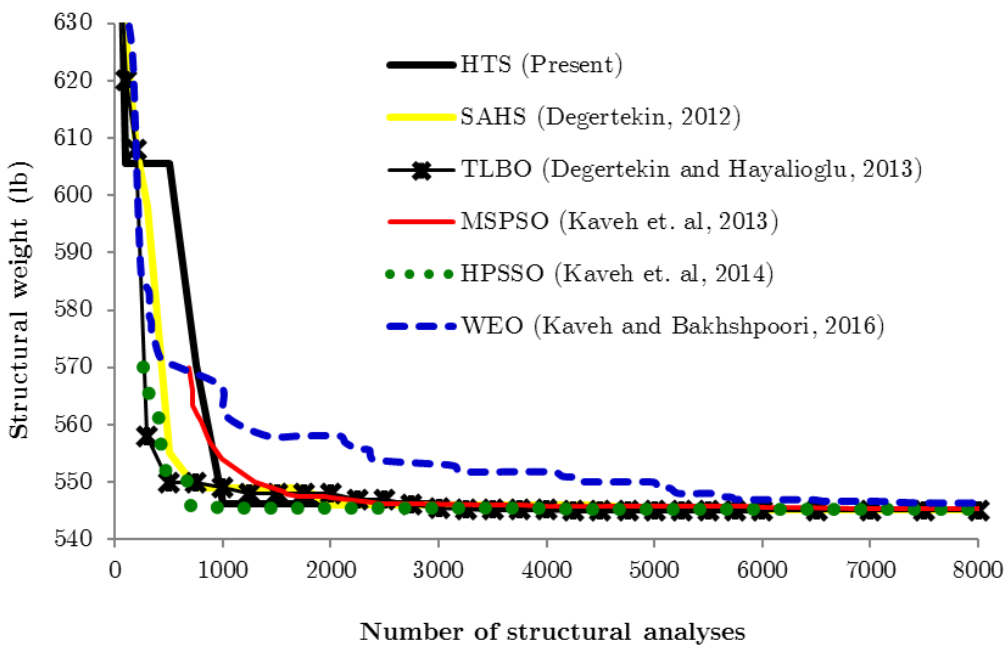

Figure 2: Comparison of convergence curves corresponding to the best optimization run of each metaheuristic algorithm for the 25-bar truss problem.

5.2 Spatial 72-Bar Truss Structure

The spatial 72-bar truss structure shown in Fig. 3 is chosen as second test problem. Material properties are the same as for the previous example. Cross-sectional areas of members are divided into 16 groups because of structural symmetry: (1) A_1-A_4, (2) A_5-A_{12}, (3) A_{13}-A_{16}, (4) A_{17}-A_{18}, (5) A_{19}-A_{22}, (6) A_{23}-A_{30}, (7) A_{31}-A_{34}, (8) A_{35}-A_{36}, (9) A_{37}-A_{40}, (10) A_{41}-A_{48}, (11) A_{49}-A_{52}, (12) A_{53}-A_{54}, (13) A_{55}-A_{58}, (14) A_{59}-A_{66}, (15) A_{67}-A_{70}, (16) A_{71}-A_{72}.

The structure is subject to the two independent loading conditions listed in Table 4. The allowable stress for all members is ±25 ksi in tension/compression while the maximum displacement of the four top nodes of the structure in all coordinate directions is ±0.25 in. The lower limit of cross-sectional areas is set as 0.1 in^2 and 0.01 in^2, respectively, for Case 1 and Case 2.

Tables 5 and 6 present, respectively, for Case 1 and Case 2, the results obtained by the HTS algorithm and the other metaheuristic optimization methods considered in this study. In Case 1, HTS found the optimum design weighing 379.73 lb after 13166 structural analyses, hence much before the limit number of 20000 analyses set for this test problem (see Table 5). This weight is only 0.1 lb heavier than the lightest weight found by TLBO (Degertekin and Hayalioglu, 2013). Although FPA (Bekdaş et. al, 2015)converged to the very small structural weight of 379.095 lb, it should actually be considered the worst algorithm because the corresponding optimized design violated problem constraints.

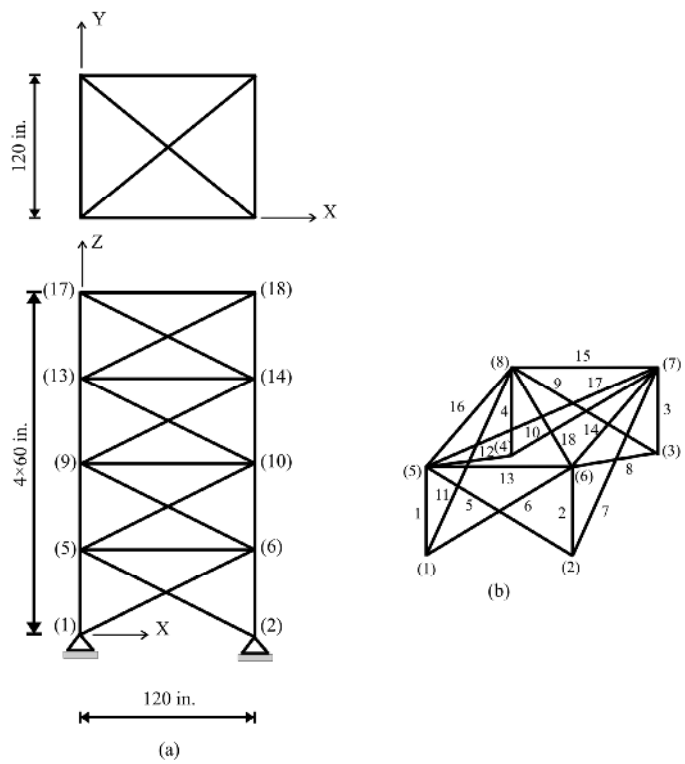

Figure 3: Schematic of the spatial 72-bar truss structure.

	Condition 1			Condition 2		
Node	F_x	F_y	F_z	F_x	F_y	F_z
17	5.0	5.0	-5.0	0.0	0.0	-5.0
18	0.0	0.0	0.0	0.0	0.0	-5.0
19	0.0	0.0	0.0	0.0	0.0	-5.0
20	0.0	0.0	0.0	0.0	0.0	-5.0

Note: loads are in kips

Table 4: Loading conditions for the spatial 72-bar truss.

Design variables A_i (in^2)	HBB-BC (Kaveh and Talatahari, 2009b)	SAHS (Degertekin, 2012)	TLBO (Degertekin and Hayalioglu, 2013)	FPA (Bekdaş et. al, 2015)	HTS This study
A_1-A_4	1.9042	1.860	1.8807	1.8758	1.9001
A_5-A_{12}	0.5162	0.521	0.5142	0.5160	0.5131
A_{13}-A_{16}	0.100	0.100	0.1000	0.1000	0.1000
A_{17}-A_{18}	0.100	0.100	0.1000	0.1000	0.1000
A_{19}-A_{22}	1.2582	1.293	1.2711	1.2993	1.2456
A_{23}-A_{30}	0.5035	0.511	0.5151	0.5246	0.5080
A_{31}-A_{34}	0.100	0.100	0.1000	0.1001	0.1000
A_{35}-A_{36}	0.100	0.100	0.1000	0.1000	0.1000
A_{37}-A_{40}	0.5178	0.499	0.5317	0.4971	0.5550
A_{41}-A_{48}	0.5214	0.501	0.5134	0.5089	0.5227
A_{49}-A_{52}	0.100	0.100	0.1000	0.1000	0.1000
A_{53}-A_{54}	0.1007	0.100	0.1000	0.1000	0.1000
A_{55}-A_{58}	0.1566	0.168	0.1565	0.1575	0.1566
A_{59}-A_{66}	0.5421	0.584	0.5429	0.5329	0.5407
A_{67}-A_{70}	0.4132	0.433	0.4081	0.4089	0.4084
A_{71}-A_{72}	0.5756	0.520	0.5733	0.5731	0.5669
Weight (lb)	379.66	380.62	379.632	379.095	379.73
Average weight (lb)	381.85	382.42	379.759	379.534	382.26
Std dev (lb)	1.201	1.38	0.149	0.272	1.94
Constr. tol (%)	None	None	None	0.2039	None
No. struct. analyses	13200	13742	21542	9029	13166

Table 5: Optimization results for Case 1 of the 72-bar truss problem.

HTS was slightly less robust than SAHS (Degertekin, 2012) and HBB-BC (Kaveh and Talatahari, 2009b). TLBO (Degertekin and Hayalioglu, 2013) was the most robust algorithm overall with a standard deviation on optimized weight of only 0.149 lb. However, such a small dispersion was obtained for a considerably larger number of structural analyses than in the case of HTS (i.e. 21542 analyses vs. only 13166 required by HTS).

The present algorithm was the fastest optimizer overall as it required considerably less structural analyses than TLBO (Degertekin and Hayalioglu, 2013): only 13166 analyses (well below the limit number of 20000) vs. 18460 analyses. SAHS (Degertekin, 2012) required almost the same number of analyses as HTS (13742 vs. 13166) but converged to a 0.9 lb heavier weight than HTS.

The present algorithm found practically the same design as HBB-BC (Kaveh and Talatahari, 2009b) within the same number of structural analyses.

Convergence curves shown in Fig. 4 (yet limited to the first 14000 structural analyses that were enough for HTS to find its best weight) demonstrate the computational efficiency of HTS which required only 4500 structural analyses to find intermediate designs always better than those found by SAHS (Degertekin, 2012). The optimization histories of HTS and TLBO (Degertekin and Hayalioglu, 2013) instead coincide after 5000 structural analyses.

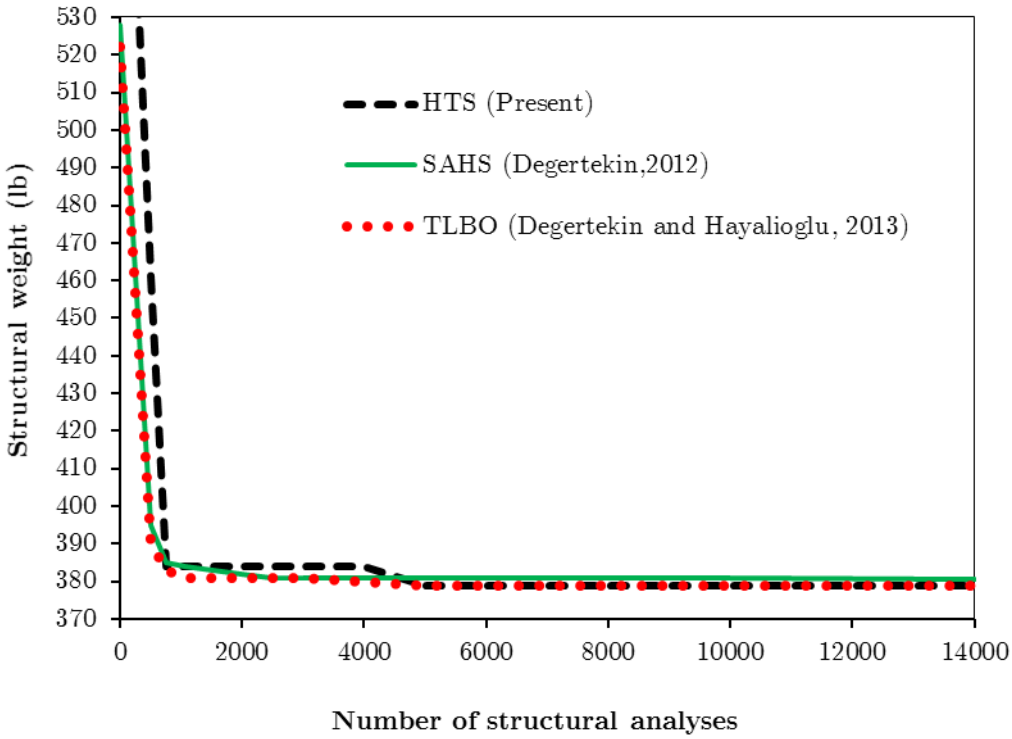

Figure 4: Comparison of convergence curves corresponding to the best optimization run of each metaheuristic algorithm for Case 1 of the 72-bar truss problem.

In Case 2, all of the compared algorithms were highly effective: in fact, optimized structural weight ranged between 363.833 lb (FFA) and 364.05 lb (SAHS), with only 0.22 lb dispersion in spite of their completely different optimization formulations (see Table 6). HTS found another very competitive optimized design weighing 363.885 lb. The results of the 30 independent optimization runs indicate the present algorithm to have the largest dispersion on optimized weight (about the same as SAHS). However, most of the other algorithms performed additional analyses to achieve more robust designs.

The number of structural analyses required by the HTS algorithm to complete the optimization process is 13592 (again well below the limit of 20000 analyses) whereas TLBO (Degertekin and Hayalioglu, 2013), MSPSO (Talatahari et. al, 2013), WEO (Kaveh and Bakhshpoori, 2016), FFA (Degertekin and Lamberti, 2013) and ABC-AP (Sonmez, 2011) were considerably slower than the

present algorithm and performed, respectively, 17954, 18400, 19860, 39404 and 400,000 structural analyses to obtain their optimum designs. Apart from SAHS (Degertekin, 2012) which required only 12852 structural analyses but converged to a slightly heavier design than HTS (i.e. 364.05 vs. 363.885 lb), the present algorithm was the second fastest optimizer overall, ranking right after HPSSO (Kaveh et. al, 2014) which required 13086 structural analyses to find the optimum design.

Design variables A_i (in^2)	ABC-AP (Sonmez, 2011)	SAHS (Degertekin, 2012)	TLBO (Degertekin and Hayalioglu, 2013)	FFA (Degertekin and Lamberti, 2013)	MSPSO (Talatahari, et. al, 2013)	HPSSO (Kaveh et. al, 2014)	WEO (Kaveh and Bakhshpoori, 2016)	This study
A_1-A_4	1.8907	1.889	1.8929	1.8945	1.9005	1.893260	1.8618	1.9000
A_5-A_{12}	0.5166	0.520	0.5160	0.5179	0.5056	0.511132	0.5206	0.5238
A_{13}-A_{16}	0.0100	0.010	0.0100	0.0100	0.0100	0.010000	0.0105	0.0100
A_{17}-A_{18}	0.0100	0.010	0.0100	0.0100	0.0100	0.010000	0.0100	0.0100
A_{19}-A_{22}	1.2968	1.289	1.29170	1.2906	1.2914	1.291227	1.2455	1.3039
A_{23}-A_{30}	0.5191	0.524	0.51759	0.5170	0.5158	0.515116	0.5177	0.5073
A_{31}-A_{34}	0.0100	0.010	0.01000	0.0100	0.0100	0.010000	0.0101	0.0100
A_{35}-A_{36}	0.0101	0.010	0.01000	0.0100	0.0100	0.010000	0.0100	0.0100
A_{37}-A_{40}	0.5208	0.539	0.52294	0.5213	0.5178	0.536082	0.5327	0.5194
A_{41}-A_{48}	0.5178	0.519	0.51925	0.5204	0.5188	0.521176	0.5109	0.5169
A_{49}-A_{52}	0.0100	0.015	0.01000	0.0100	0.0108	0.010019	0.0100	0.0100
A_{53}-A_{54}	0.1048	0.105	0.09970	0.1045	0.1165	0.110878	0.1205	0.1067
A_{55}-A_{58}	0.1675	0.167	0.16795	0.1675	0.1659	0.166691	0.1655	0.1674
A_{59}-A_{66}	0.5346	0.532	0.53594	0.5328	0.5479	0.533984	0.5397	0.5330
A_{67}-A_{70}	0.4443	0.425	0.44566	0.4455	0.4437	0.453724	0.4554	0.4536
A_{71}-A_{72}	0.5803	0.579	0.58181	0.5804	0.5619	0.574581	0.5995	0.5786
Weight (lb)	363.8392	364.0	363.841	363.833	363.900	363.8581	363.9827	363.885
Average weight (lb)	N/A	366.57	364.42	363.26	364.350	364.065	364.3536	366.03
Std dev (lb)	N/A	2.02	0.49	0.369	0.320	0.305	0.2188	2.29
Const. tol (%)	None	None	None	None	0.0864	None	None	None
No. structural analyses	400,000	12852	17954	39404	18400	13086	19860	13592

Table 6: Optimization results for Case 2 of the 72-bar truss problem.

Optimization histories for this design example are plotted in Fig. 5 for the most efficient methods compared in Table 6. The plot covers only the first 14000 structural analyses, thus including the 13592 analyses required by HTS to find its best design. It can be seen from the figure that convergence rate of HTS is as good as those of the other algorithms, especially if the best agent included in the HTS initial population yield a larger structural weight than for other optimizers (for example, TLBO and SAHS).

WEO (Kaveh and Bakhshpoori, 2016) and MSPSO (Talatahari et. al, 2013) were definitely the slowest optimizers: this is consistent with the very large numbers of structural analyses required by these algorithms (see Table 6). Convergence curves of HTS and HPSSO (Kaveh et. al, 2014) crossed each other between 600 and 1000 structural analyses. In particular, HPSSO (Bekdaş et. al, 2015) started the optimization process from a larger structural weight than HTS but it soon recovered this gap and generated intermediate designs better than those of HTS. The present algorithm returned to produce better designs between 1800 and 5000 structural analyses, and convergence curves of HTS and HPSSO (Kaveh et al, 2014) practically coincided since that point.

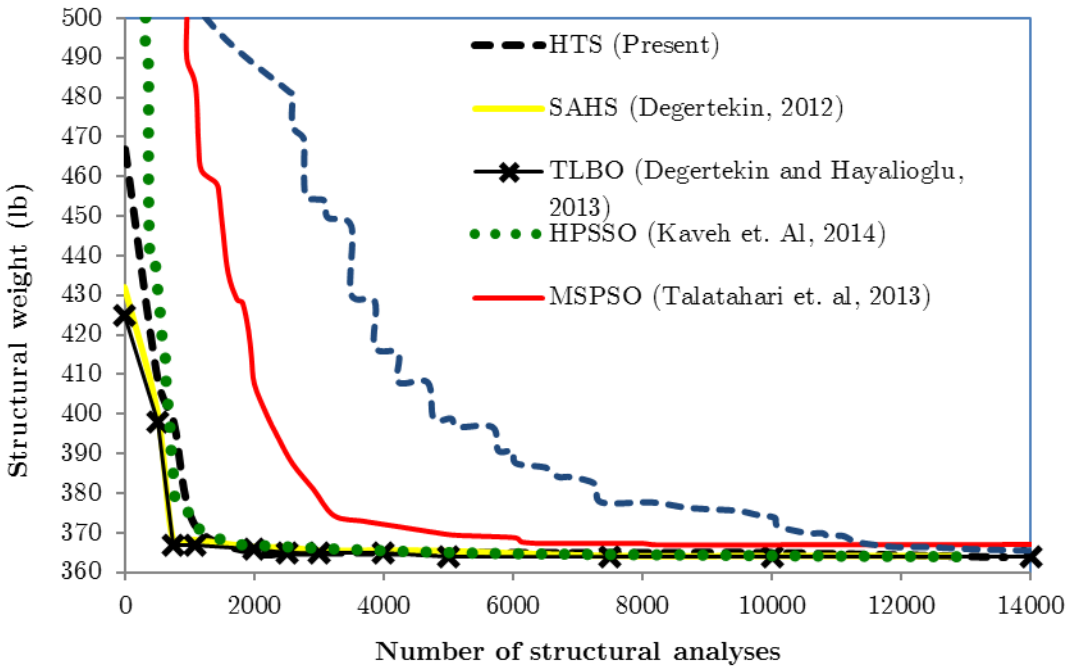

Figure 5: Comparison of convergence curves corresponding to the best optimization run of each metaheuristic algorithm for Case 2 of the 72-bar truss problem.

5.3 Planar 200-Bar Truss

The last test problem considered in this study is the weight minimization of the planar 200-bar truss structure shown in Fig. 6. This test case has often been taken as an example of average-scale optimization. The structure is made of steel: Young's modulus is 30 Msi while mass density is 0.283 lb/in^3. Because of structural symmetry, elements are divided in 29 groups as indicated in Table 7.

Three independent loading conditions act on the structure: (1) 1.0 kip acting in the positive X-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62 and 71; (2) 10.0 kips acting in the negative Y-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28,29 30, 31, 32, 33,

34,36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74 and 75; (3) loading conditions (1) and (2) acting together.

The structure must be designed against stress constraints: the allowable stress in tension/compression is ±10 ksi. No constraints on nodal displacements are defined for this optimization problem. Cross-sectional areas of elements must be greater than 0.1 in^2.

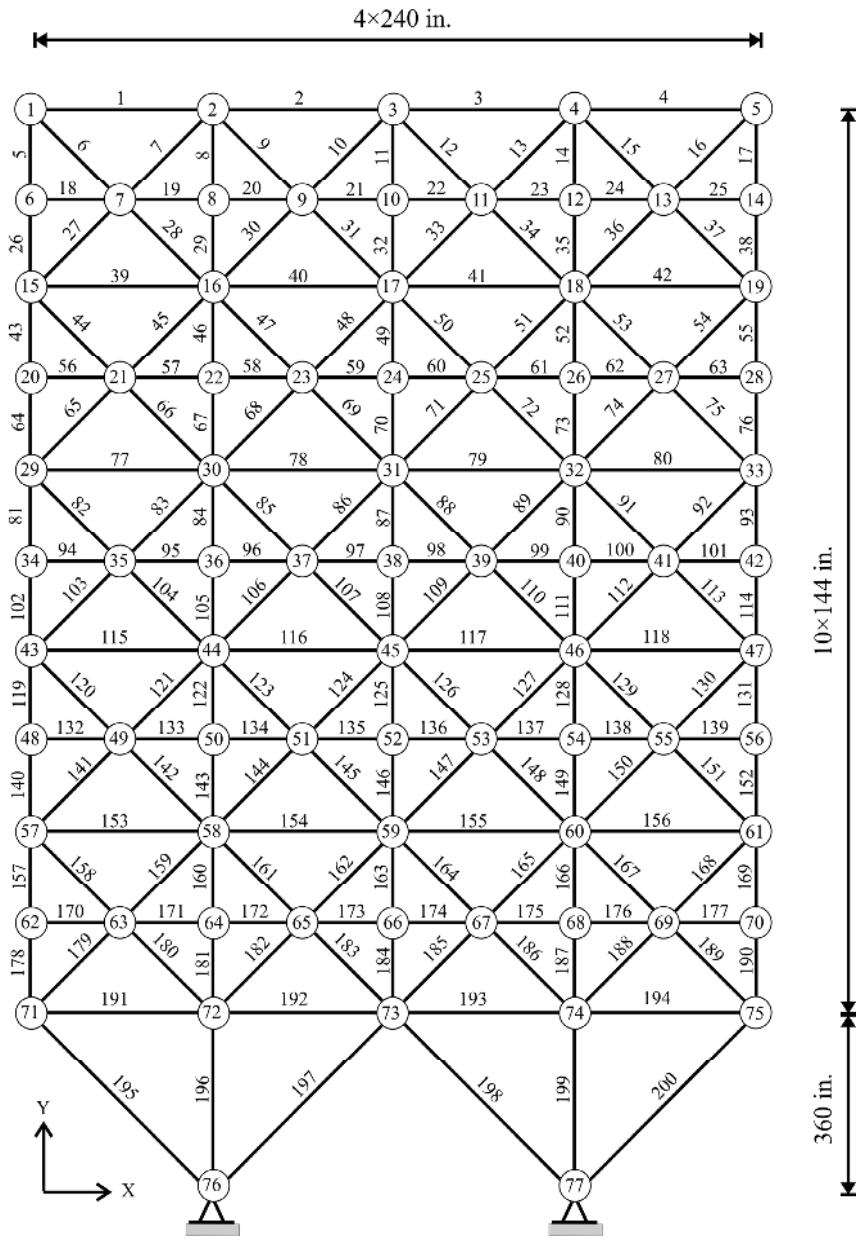

Figure 6: Schematic of the planar 200-bar truss structure.

Design variables	Member number	Design variables	Member number
1	1,2,3,4	16	82,83,85,86,88,89,91,92, 103,104,106,107,109,110,112,113
2	5,8,11,14,17	17	115,116,117,118
3	19,20,21,22,23,24	18	119,122,125,128,131
4	18,25,56,63,94,101,132, 139,170,177	19	133,134,135,136,137,138
5	26,29,32,35,38	20	140,143,146,149,152
6	6,7,9,10,12,13,15,16,27, 28,30,31,33,34,36,37	21	120,121,123,124,126,127,129, 130,141,142,144,145,147,148, 150,151
7	39,40,41,42	22	153,154,155,156
8	43,46,49,52,55	23	157,160,163,166,169
9	57,58,59,60,61,62	24	171,172,173174,175,176
10	64,67,70,73,76	25	178,181,184,187,190
11	44,45,47,48,50,51,53,54, 65,66,68,69,71,72,74,75	26	158,159,161,162,164,165,167, 168,179,180,182,183,185,186, 188,189
12	77,78,79,80	27	191,192,193,194
13	81,84,87,90,93	28	195,197,198,200
14	95,96,97,98,99,100	29	196,199
15	102,105,108,111,114		

Table 7: Member grouping and sizing variables defined in the 200-bar truss problem.

Table 8 compares the optimization results for HTS, CMLPSA (Lamberti, 2008), ABC-AP (Sonmez, 2011), SAHS (Degertekin, 2012), TLBO (Degertekin and Hayalioglu, 2013), FFA (Degertekin and Lamberti, 2013), HPSSO (Kaveh et. al, 2014), FPA (Bekdaş et al, 2015) and WEO (Kaveh and Bakhshpoori, 2016). It can be seen that the present algorithm obtained the third lowest weight (25517.31 lb), only 0.12% heavier than the designs optimized by TLBO and SAHS (i.e. 25488.15 and 25491.9 lb). It should be noted that the best weight quoted in literature for this problem is 25156.5 lb, achieved by the HPSACO algorithm (Kaveh and Talatahari, 2009c) by hybridizing harmony search, particle swarm and ant colony. However, since stress constraints were violated by 9.97%, such a design should not be considered very indicative. The second best weight quoted in literature for metaheuristic algorithms is 25445.63 lb, achieved by the CMLPSA algorithm (Lamberti, 2008). As the scaled weight of CMLPSA to recover the 0.071% stress constraint violation of its optimized design is 25463.7 lb, the present HTS algorithm designed a structure only 0.21% heavier than that designed by CMLPSA.

The present algorithm is considerably more robust than HPSSO (Kaveh et. al, 2014) and WEO (Kaveh and Bakhshpoori, 2016) and enough more robust than SAHS (Degertekin, 2012). The other algorithms are characterized by a smaller standard deviation but also by a higher computational cost or constraint violation.

Design variables A_i (in^2)	CMLPSA (Lamberti, 2008)	ABC-AP (Sonmez, 2011)	SAHS (Degertekin, 2012)	TLBO (Degertekin and Hayalioglu, 2013)	FFA (Degertekin and Lamberti, 2013)	HPSSO (Kaveh et al, 2014)	FPA (Bekdaş et al, 2015)	WEO (Kaveh and Bakshspoori, 2016)	HTS This study
1	0.1468	0.1039	0.154	0.146	0.155	0.1213	0.1425	0.1144	0.151
2	0.9400	0.9463	0.941	0.941	0.941	0.9426	0.9637	0.9443	0.948
3	0.1000	0.1037	0.100	0.100	0.100	0.1220	0.1005	0.1310	0.101
4	0.1000	0.1126	0.100	0.101	0.100	0.1000	0.1000	0.1016	0.103
5	1.9400	1.9520	1.942	1.941	1.944	2.0143	1.9514	2.0353	1.947
6	0.2962	0.293	0.301	0.296	0.299	0.2800	0.2957	0.3126	0.297
7	0.1000	0.1064	0.100	0.100	0.105	0.1589	0.1156	0.1679	0.100
8	3.1042	3.1249	3.108	3.121	3.109	3.0666	3.1133	3.1541	3.115
9	0.1000	0.1077	0.100	0.100	0.100	0.1002	0.1006	0.1003	0.102
10	4.1042	4.1286	4.106	4.173	4.111	4.0418	4.1100	4.1005	4.202
11	0.4034	0.4250	0.409	0.401	0.406	0.4142	0.4165	0.4350	0.404
12	0.1912	0.1046	0.191	0.181	0.193	0.4852	0.1843	0.1148	0.171
13	5.4284	5.4803	5.428	5.423	5.449	5.4196	5.4567	5.3823	5.430
14	0.1000	0.1060	0.100	0.100	0.100	0.1000	0.1000	0.1607	0.101
15	6.4284	6.4853	6.427	6.422	6.439	6.3749	6.4559	6.4152	6.426
16	0.5734	0.5600	0.581	0.571	0.579	0.6813	0.5800	0.5629	0.568
17	0.1327	0.1825	0.151	0.156	0.134	0.1576	0.1547	0.4010	0.159
18	7.9717	8.0445	7.973	7.958	7.993	8.1447	8.0132	7.9735	7.957
19	0.1000	0.1026	0.100	0.100	0.100	0.1000	0.1000	0.1092	0.100
20	8.9717	9.0334	8.974	8.958	8.983	9.0920	9.0135	9.0155	8.957
21	0.7049	0.7844	0.719	0.720	0.708	0.7462	0.7391	0.8628	0.721
22	0.4196	0.7506	0.422	0.478	0.421	0.2114	0.7870	0.2220	0.482
23	10.8636	11.3057	10.892	10.897	10.871	10.9587	11.1795	11.0254	10.901
24	0.1000	0.2208	0.100	0.100	0.100	0.1000	0.1462	0.1397	0.103
25	11.8606	12.2730	11.887	11.897	11.883	11.9832	12.1799	12.0340	11.942
26	1.0339	1.4055	1.040	1.080	1.037	0.9241	1.3424	1.0043	1.078
27	6.6818	5.1600	6.646	6.462	6.689	6.7676	5.4844	6.5762	6.456
28	10.8113	9.9930	10.804	10.799	10.825	10.9639	10.1372	10.7265	10.819
29	13.8404	14.70144	13.870	13.922	13.847	13.8186	14.5262	13.9666	13.918
Weight (lb)	25445.63	25533.79	25491.9	25488.15	25497.8	25698.85	25521.81	25674.83	25517.31
Aver. weight (lb)	25446.03	N/A	25610.2	25533.14	25560.12	28386.72	25543.51	26613.45	25565.33
Std dev (lb)	N/A	N/A	141.85	27.44	63.95	2403	18.13	702.80	114.33
Constr. tol (%)	0.071	13.136	None	None	None	None	0.169	None	None
No. struct Analyses	9650	1,450,000	19670	28059	95680	14406	10685	19410	19661

Table 8: Optimization results for the 200-bar truss problem.

HTS reached its optimum design after 19661 structural analyses (once again much before the limit number of 25000 analyses set for this test problem) whereas ABC-AP (Sonmez, 2011), FFA (Degertekin and Lamberti, 2013) and TLBO (Degertekin and Hayalioglu, 2013), respectively, required 1,450,000, 95680 and 28059 analyses; HTS, SAHS (Degertekin, 2012) and WEO (Kaveh and

Bakhshpoori, 2016) required practically the same number of analyses. HPSSO (Kaveh et. al, 2014) was 27% faster than HTS but its optimized weight is 0.71% heavier than that found by HTS. The optimum design found by HTS strictly satisfies stress constraints while ABC-AP (Sonmez, 2011) and FPA (Bekdaş et. al, 2015) converged to unfeasible designs. Comparison with CMPLSA (Lamberti, 2008) is not indicative as this algorithm utilized gradient information that are explicitly available in sizing optimization of truss structures while HTS carries out a strictly metaheuristic search.

Figure 7 compares the optimization histories of HTS, SAHS (Degertekin, 2012), TLBO (Degertekin and Hayalioglu, 2013), HPSSO (Kaveh et. al, 2014) and FFA (Degertekin and Lamberti, 2013). Convergence curves cover only the first 24000 structural analyses of the search process in order to highlight the following facts: (i) HTS and SAHS (Degertekin, 2012) successfully terminate their search well before reaching the limit number of analyses; (ii) FFA (Degertekin and Lamberti, 2013) and HPSSO (Kaveh et. al, 2014) approach optimum design more slowly and HPPSO (Kaveh et. al, 2014) is trapped in a local optimum. It should be noted that HTS, SAHS (Degertekin, 2012) and TLBO (Degertekin and Hayalioglu, 2013) started their best runs from a better population than HPSSO (Kaveh et. al, 2014) and FFA (Degertekin and Lamberti, 2013): in fact, the most efficient initial designs for these algorithms were up to two times heavier than those selected by the other optimizers (i.e. about 114000 lb for HPSSO (Kaveh et. al, 2014) and 94000 lb for FFA (Degertekin and Lamberti, 2013) vs. 50000-60000 lb for HTS, SAHS (Degertekin, 2012)and TLBO (Degertekin and Hayalioglu, 2013). In spite of this, optimum designs finally obtained by each algorithm differ by only 0.83%. However, while HPSSO (Kaveh et. al, 2014) prematurely converged to a local optimum, structural weights of all intermediate designs found by HTS, SAHS (Degertekin, 2012), TLBO (Degertekin and Hayalioglu, 2013) and FFA (Degertekin and Lamberti, 2013) differed by less than 0.5% after only 17000 structural analyses.

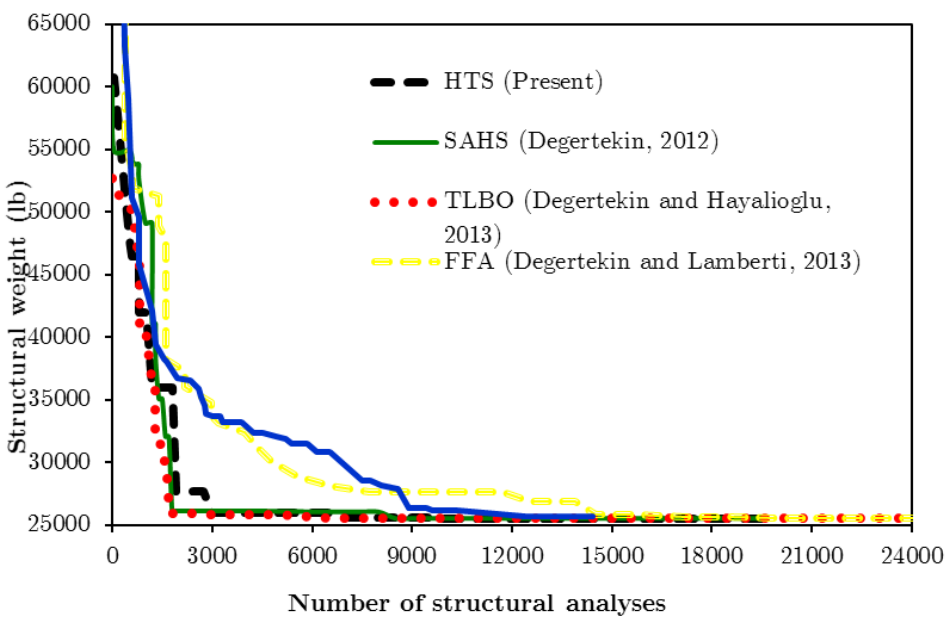

Figure 7: Comparison of convergence curves corresponding to the best run of each metaheuristic algorithm for the planar 200-bar truss problem.

6 CONCLUDING REMARKS

The heat transfer algorithm (HTS), a metaheuristic optimization method developed very recently (2015), was applied for the first time ever to sizing optimization of truss structures. The three design examples discussed in the article demonstrate that HTS often obtained better designs than those found by other state-of-the-art metaheuristic optimization methods. The convergence capability of HTS is also as good as for the other methods. The presented data on average weight and standard deviation of optimized weight obtained from thirty independent runs prove the robustness of HTS which hence has the ability to find the global optimum or a nearly global optimum design. This is due to the inherent ability of HTS to keep a very good balance between exploration and exploitation mechanisms throughout the optimization process. In fact, exploration and exploitation are alternatively activated by the sequence of conduction, convection and radiation phases as well as they alternate in each specific phase. The proposed HTS algorithm is very general and could be applied to sizing and layout optimization of other structural systems like frames and domes in future investigations.

References

Bekdaş G., Nigdeli S.M., Yang X.S .(2015). Sizing optimization of truss structures using flower pollination algorithm. Appl Soft Comput. doi: 10.1016/j.asoc.2015.08.037

Camp C.V., Farshchin M. (2014). Design of space trusses using modified teaching-learning based optimization. Eng Struct. doi: 10.1016/j.engstruct.2014.01.020

Çengel Y.A. (2008). Introduction to Thermodynamics and Heat Transfer. 2nd Edition. McGraw-Hill

Degertekin S.O. (2012). An improved harmony search algorithms for sizing optimization of truss structures. Comput Struct. doi: 10.1016/j.compstruc.2011.10.022

Degertekin S.O., Hayalioglu M.S. (2013). Sizing truss structures using teaching-learning-based optimization. Comput Struct. doi: 10.1016/j.compstruc.2012.12.011.

Degertekin S.O., Lamberti L. (2013). Sizing optimization of truss structures using the firefly algorithm. Proceedings of the Fourteenth International Conference on Civil, Structural and Environmental Engineering Computing, Paper 229, Civil-Comp Press, UK.

Erol O.K., Eksin I. (2006). A new optimization method: big bang-big crunch. Adv Eng Softw, doi: 10.1016/j.advengsoft.2005.04.005

Geem Z.W., Kim J.H., Loganathan G.V. (2001). A new heuristic optimization algorithm: harmony search. Simulation 76, 60-68.

Hollman J.P. (2010). Heat Transfer 10th Edition. Mc Graw-Hill, New York

Karaboga D. (2005). An idea based on honey bee swarm for numerical optimization. technical report-TR06 Erciyes University Engineering Faculty Computer Engineering Department, Kayseri (Turkey).

Kaveh A, Bakhshpoori T (2016) A new metaheuristic for continuous structural optimization: water evaporation optimization. Struct Multidiscip Optim. doi: 10.1007/s00158-015-1396-8

Kaveh A, Bakhshpoori T, Afshari E (2014) An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct. doi: 10.1016/j.compstruc.2014.07.012

Kaveh A. (2014). Advances in Metaheuristic Algorithms for Optimal Design of Structures, Springer International Publishing, Switzerland.

Kaveh A., Talatahari S. (2009a). A particle swarm ant colony optimization for truss structures with discrete variables. J Constr Steel Res. doi: 10.1016/j.jcsr.2009.04.021

Kaveh A., Talatahari S. (2009b). Size optimization of space trusses using Big Bang-Big Crunch algorithm. Comput Struct. doi: 10.1016/j.compstruc.2009.04.011

Kaveh A., Talatahari S. (2009c). Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct. doi: 10.1016/j.compstruc.2009.01.003

Kaveh A., Talatahari S. (2010). A novel heuristic optimization method: charged system search. Acta Mech. doi: 10.1007/s00707-009-0270-4

Kennedy J., Eberhart R. (1995). Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948.

Lamberti L, Pappalettere C (2011) Metaheuristic design optimization of skeletal structures: a review. Computational Technology Reviews 4, 1-32.

Lamberti L. (2008). An efficient simulated annealing algorithm for design optimization of truss structures. Comput Struct. doi: 10.1016/j.compstruc.2008.02.004

Neshat M., Sepidnam G., Sargolzaei M. (2013). Swallow swarm optimization algorithm: a new method to optimization. Neural Comput Appl. doi: 10.1007/s00521-012-0939-9

Patel V.K., Savsani V.J. (2015). Heat transfer search (HTS): a novel optimization algorithm. Inform Sciences. doi: 10.1016/j.ins.2015.06.044

Perez R.E., Behdinan K. (2007). Particle swarm approach for structural design optimization, Comput Struct. doi: 10.1016/j.compstruc.2006.10.013

Rao R.V., Savsani V.J., Vakharia D.P. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des. doi: 10.1016/j.cad.2010.12.015

Saka M.P., Dogan E. (2012). Recent developments in metaheuristic algorithms: a review. Computational Technology Reviews 5, 31-78.

Sonmez M. (2011). Artificial bee colony algorithm for optimization of truss structures. Appl Soft Comput. doi: 10.1016/j.asoc.2010.09.003

Talatahari S., Kheirollahi M., Farahmandpour C., Gandomi A.H. (2013). A multi-stage particle swarm for optimum design of truss structures. Neural Comput & Applic. doi: 10.1007/s00521-012-1072-5

von Böckh P., Wetzel T. (2012) Heat Transfer. Springer-Verlag

Yang X.S. (2010). Engineering Optimization An Introduction with Metaheuristic Applications. John Wiley & Sons, UK

Yang X.S. (2012). Flower pollination algorithm for global optimization. In: Unconventional Computation and Natural Computation, pp. 240-249.