Steel and Composite Structures, Vol. 8, No. 6 (2008) 475-490

DOI: http://dx.doi.org/10.12989/scs.2008.8.6.475

A hybrid tabu-simulated annealing heuristic algorithm for optimum design of steel frames

S.O. Degertekin* and M.S. Hayalioglu

Department of Civil Engineering, Dicle University, 21280, Diyarbakir, Turkey

M. Ulker

Department of Civil Engineering, Firat University, 23119, Elazig, Turkey

(Received March 6, 2008, Accepted September 19, 2008)

Abstract. A hybrid tabu-simulated annealing algorithm is proposed for the optimum design of steel frames. The special character of the hybrid algorithm is that it exploits both tabu search and simulated annealing algorithms simultaneously to obtain near optimum. The objective of optimum design problem is to minimize the weight of steel frames under the actual design constraints of AISC-LRFD specification. The performance and reliability of the hybrid algorithm were compared with other algorithms such as tabu search, simulated annealing and genetic algorithm using benchmark examples. The comparisons showed that the hybrid algorithm results in lighter structures for the presented examples.

Keywords : optimum design; steel frames; tabu search; simulated annealing; hybrid heuristic algorithm.

1. Introduction

Many researchers have been interested in heuristic search algorithms such as genetic algorithms (GAs), tabu search (TS) and simulated annealing (SA) for the last two decades. The reason for this is that they could find optimal or near optimal solutions by means of their powerful tools. Near optimal solutions (sometimes optimal) determined by heuristic approach is more preferable and acceptable in practice because they can be obtained more efficiently (Chan *et al*. 2005).

Broadly speaking, all heuristic search algorithms imitate a natural phenomenon. GAs are based on evolution theory of Darwin's. Initially, they were proposed by Holland (1975). The main principle of GAs is the survival of robust ones and the elimination of the others in a population. They have been applied widely to optimum design of steel frames (Jenkins 1992, Camp *et al*. 1998, Pezeshk *et al*. 2000, Hayalioglu 2000,2001, Toropov and Mahfouz 2001, Kameshki and Saka 2001,2003, Hayalioglu and Degertekin 2005, Liu *et al*. 2006, Degertekin 2007, Degertekin *et al*. 2008). SA mimics the annealing process in solids. It was developed by Metropolis *et al*. (1953) and proposed by Kirkpatrick *et al*. (1983) for optimization problems. It has also been applied to optimum design of steel frames (Balling 1991, Huang and Arora 1997, Park and Sung 2002, Hayalioglu and Degertekin 2007, Degertekin 2007). TS is

*Corresponding Author, E-mail: sozgur@dicle.edu.tr

another heuristic algorithm which imitates the human memory process. It was developed by Glover (1989, 1990). TS has an artificial memory and saves information about recent search moves using tabu list which forbids recently made moves. Therefore; the probability of becoming entrapped into local optima is prevented. Optimum design of steel frames using TS has been studied relatively less than GAs and SA (Kargahi *et al*. 2006, Kargahi and Anderson 2006, Hayalioglu and Degertekin 2007, Degertekin *et al*. 2008).

In recent years, hybrid search algorithms have emerged in optimization problems. They combine together superiority of different optimization methods. The aim of hybridization is to improve the performance of algorithms that are based on a single method (Mantawy *et al*. 1997). Hybrid search algorithms are getting more and more popular. One of them is hybrid tabu-simulated annealing algorithm (HTS). The applications of HTS could be summarized as follows: unit commitment problem (Mantawy *et al*. 1997), jointly solving the group scheduling and machining speed selection (Zolfaghari and Liang 1999), optimal power flow problems (Ongsakul and Bhasaputra 2002), machine loading problem of flexible manufacturing system (Swarnkar and Tiwari 2004), multi-constraint product mix decision problem (Mishra *et al*. 2005), problem of packing circles into a larger containing circle (Zhang and Deng 2005), optimal stacking sequence design of laminated composite structures (Rao and Arvind 2007), warehouse-scheduling problems (Chan and Kumar 2008).

The main goal of this paper is to introduce/apply a HTS algorithm for optimum design of steel frames. The objective of optimum design problem is to minimize the weight of steel frames under the actual design constraints of code specifications of AISC-LRFD (2001). The performance and reliability of HTS were proved using three steel frames which exist in literature.

2. The formulations of optimum design problem

The objective of optimum design problem is to minimize the weight of steel frames and it could be defined as:

$$\text{Minimize } W(x) = \sum_{k=1}^{ng} A_k \sum_{i=1}^{mk} \rho_i L_i \tag{1}$$

where mk is the total numbers of members in group k, ρ_i and L_i are density and length of member i, A_k is cross-sectional area of member group k, and ng is total numbers of groups in the frame. A design variable in the optimization problem designates a member group in a steel design. Design variables are selected from a section list and each of section is represented by a sequence number in that list. In this study, the design examples were taken from authors' previous works (Degertekin 2007, Hayalioglu and Degertekin 2007, Degertekin *et al*. 2008) for making comparisons. Therefore, the same optimum design formulations as the ones of those articles are used herein.

The displacement constraints are (Degertekin *et al*. 2008):

$$\lambda_{jl}(x) = \frac{\delta_{jl}}{\delta_{ju}} - 1 \le 0, \quad j = 1, \ldots\ldots, m, \qquad l = 1, \ldots\ldots nl \tag{2}$$

$$\lambda_{jil}(x) = \frac{\Delta_{jil}}{\Delta_{ju}} - 1 \le 0, \quad j = 1, \ldots\ldots, ns, \qquad i = 1, \ldots\ldots, nsc, \quad l = 1, \ldots\ldots nl \tag{3}$$

where δ_{jl} is the displacement of the *j-th* degree of freedom due to loading condition *l*, δ_{ju} is its upper bound, *m* is the number of restricted displacements, *nl* is the total number of loading conditions, Δ_{jil} is interstorey drift of *i*-th column in the *j*-th storey due to loading condition *l*, Δ_{ju} is its limit, *ns* is the number of storeys in the frame, *nsc* is the number of columns in a storey.

The strength constraints taken from AISC-LRFD (2001) are expressed in the following equations. For members subject to bending moment and axial force:

for $\frac{P_u}{\phi P_n} \geq 0.2$,

$$\lambda_{il}(x) = \left(\frac{P_u}{\phi P_n}\right)_{il} + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) - 1.0 \leq 0 \qquad i = 1, \ldots, nm, \quad l = 1, \ldots\ldots, nl \tag{4}$$

for $\frac{P_u}{\phi P_n} < 0.2$,

$$\lambda_{il}(x) = \left(\frac{P_u}{2\phi P_n}\right)_{il} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) - 1.0 \leq 0 \qquad i = 1, \ldots, nm, \quad l = 1, \ldots\ldots, nl \tag{5}$$

where *nm* is total number of members in the frame, P_u = required axial strength (compression or tension), P_{n} = nominal axial strength (compression or tension), M_{ux} = required flexural strengths about the major axis (for two-dimensional frames, $M_{uy} = 0$), M_{uy} = required flexural strengths about the minor axis, M_{nx} = nominal flexural strength about the major axis, M_{ny} = nominal flexural strength about the minor axis, $\phi = \phi_c$ = resistance factor for compression (equal to 0.85), $\phi = \phi_t$ = resistance factor for tension (equal to 0.90), ϕ_b = flexural resistance factor (equal to 0.90).
The size constraints are given as:

$$\lambda_n(x) = \frac{d_{un}}{d_{bn}} - 1.0 \leq 0, \qquad n = 1, \ldots\ldots, ncl \tag{6}$$

where d_{un} and d_{bn} are depths of steel sections selected for upper and lower floor columns, *ncl* is the total number of columns in the frame except the ones at the bottom floor.

The unconstrained objective function *φ(x)* is then written as (Degertekin *et al*. 2008):

$$\varphi(x) = W(x)\left[1 + C\left(\sum_{j=1}^{m}\sum_{l=1}^{nl} v_{jl} + \sum_{j=1}^{ns}\sum_{i=1}^{nsc}\sum_{l=1}^{nl} v_{jil} + \sum_{i=1}^{nm}\sum_{l=1}^{nl} v_{il} + \sum_{n=1}^{ncl} v_n\right)\right] \tag{7}$$

where C is a penalty constant, v_{jl}, v_{jil}, v_{il} and v_n are violation coefficients which are calculated as

$$\text{if } \lambda_{il}(x) > 0 \quad \text{then} \quad v_{jl} = \lambda_{jl}(x)$$

$$\text{if } \lambda_{jl}(x) \leq 0 \quad \text{then} \quad v_{jl} = 0$$

$$\begin{array}{lll} \text{if } \lambda_{jil}(x) > 0 & \text{then} & v_{jil} = \lambda_{jil}(x) \\ \text{if } \lambda_{jil}(x) \leq 0 & \text{then} & v_{jil} = 0 \\ \text{if } \lambda_{il}(x) > 0 & \text{then} & v_{il} = \lambda_{il}(x) \\ \text{if } \lambda_{il}(x) \leq 0 & \text{then} & v_{il} = 0 \\ \text{if } \lambda_{n}(x) > 0 & \text{then} & v_{n} = \lambda_{n}(x) \\ \text{if } \lambda_{n}(x) \leq 0 & \text{then} & v_{n} = 0 \end{array} \tag{8}$$

3. Tabu search

TS is a powerful heuristic search algorithm which is based on the human memory process and uses an iterative neighbourhood search procedure. It uses a move operator in order to generate neighbour designs of the current one.

TS has two distinctive features which are tabu list and aspiration criterion. Tabu list forbids the recently made moves in order to prevent turning back to the old designs. It also leads the search process to different areas in the design space. Aspiration criterion allows the search to ignore the tabu status of a design if it is the best design that has been obtained so far. Therefore, aspiration criterion increases the flexibility of the algorithm through an intensification of the search in neighbourhoods with good solutions (Dhingra and Bennage 1995).

Although TS uses powerful tools such as tabu list and aspiration criterion to obtain global optimum, it has a drawback. This drawback may emerge if the search turns back to a recently visited solution with a move that is not in tabu list. Therefore, a loop is come out. It may be averted by using longer tabu list or larger neighbourhood depth, but longer tabu list may constrain the search and larger neighbourhood depth causes longer computational effort (Zolfaghari and Liang 1999). Another way of preventing the TS to get trapped in a loop is using frequency-based memory (Glover and Laguna, 1997).

4. Simulated Annealing

SA bases on an analogy between the annealing of solids and searching the solutions to optimization problems. It has a distinctive feature to prevent local optima which is known as acceptance probability. SA not only accepts lighter neighbour designs but also accepts heavier ones with an acceptance probability. Although SA has superior facilities such as random perturbation and acceptance probability, it generally requires more iterations and longer computational effort (Swarnkar and Tiwari 2004). Another drawback of SA when compared with TS is that it does not have memory, hence it is possible to return to some recently visited solutions (Zolfaghari and Liang 1999).

A hybrid tabu-simulated annealing algorithm (HTS) could be proposed which exploits the advantages of both TS and SA algorithms simultaneously. The details of HTS will be explained in the following section.

5. A hybrid tabu-simulated annealing algorithm

Although TS has short-term memory and aspiration criterion, there is still a possibility of cycling as explained in the Section 3. Moreover, SA is able to escape from local optima using its acceptance probability, but it has not memory facility in contrast to TS. Hence, SA is prone to turn back recently visited solutions.

If SA uses tabu list in order to prevent the searching process from turning back to recently visited solutions, the performance of SA can be significantly improved (Zolfaghari and Liang 1999). The drawbacks existed in both TS and SA could be rectified by using a hybrid tabu-simulated annealing algorithm. Actually, HTS combines stochastic nature of SA with deterministic approach of TS. Acceptance probability, tabu list and aspiration criterion are used at the same time. The proposed algorithm basically uses the SA algorithm which was proposed by author's previous work (Degertekin 2007), but it was supplemented with powerful TS tools which are tabu list and aspiration criterion. The details of HTS proposed in this study will be explained in the following subsections.

5.1 Generation of initial designs

HTS initially generate 100 different designs (i.e. steel frames) randomly. This is done in the following way:

A random integer number is generated for each member group in the frame between one and the size of the section list (i.e. the total number of sections in the list) using a random number generator. These numbers represent the sequence numbers of the sections in the list. The steel sections corresponding to these random numbers are assigned to each member group and then the frame is analyzed with these sections. The value of the unconstrained objective function of this design is calculated using Eqs. (1)-(8). The same procedures are repeated 99 times and thus 100 different initial designs are obtained. The design with the lowest value of the unconstrained objective function among the 100 designs is assigned as 'current design' and the others are eliminated. If the current design satisfies all the constraints, it is also assigned as 'current optimum design' and recorded in the aspiration list. The aim of generating different initial designs is to find a better initial design for HTS algorithm.

5.2 Neighbourhood search and tabu list

In this step, various designs are obtained by iterative neighbourhood search procedure. This procedure is applied in the following way (Hayalioglu and Degertekin 2007): A variable of current design is selected randomly and a candidate design is obtained by changing only that variable in the range of a predetermined neighbourhood depth. If the neighbourhood depth is determined as ± 3, a candidate design is obtained by exchanging the selected variable with the one within the range of three upper and three lower variables in the sequence of the list. Let us consider an initial design with three design variables: 9,48,61. These numbers represent the sequence numbers in a determined list of discrete variables. Let the second variable (number 48) be selected randomly for perturbation and -2 value is generated randomly for the neighbourhood depth ± 3. Therefore, a candidate design is generated as (9,46,61). Structural analysis and response of this design are obtained and its unconstrained objective function value is calculated using Eqs. (1)-(8). Meanwhile, the move (design variable) is recorded in a one-dimensional list called "tabu list". (46-th discrete variable is recorded in tabu list). The other design variables of the neighbour design are also checked whether they are in the tabu list or not. This design is replaced with the current design with an acceptance probability (Section 5.4) even if a design variable of it is not in the tabu list and the search process continues starting with this new current design. If all variables of a neighbourhood design are in the tabu list, it is checked whether or not it satisfies aspiration criteria which will be explained in the following subsection.

Tabu list is a one-dimensional array whose size is kept constant during the search process. For this reason, when the tabu list is filled the oldest move at the beginning of the list is dropped and a new move is put into the end of the list (Hayalioglu and Degertekin 2007).

5.3 Aspiration criteria

If a neighbour design satisfies all the constraints, it is recorded in a list with single member. This list is called “aspiration list”. The aspiration list is updated throughout the search when a better feasible design is encountered than the current one. During the search process, even if all variables of a neighbour design are in the tabu list, its tabu status is temporarily ignored providing that it is a better design than the one in the aspiration list and satisfies all the constraints. These conditions are called “aspiration criteria”. This design is accepted as new current design and also put into the aspiration list as the current optimum design. This design is rejected if it does not satisfy the aspiration criteria and it is declared as “tabu”.

5.4 Acceptance probability

If a neighbour design is not tabu, the acceptance probability of this design as a new current design is also checked. The acceptance probability A_{ij} of accepting candidate design j which has been generated from current design i is given as (Degertekin 2007)

$$A_{ij}(T_k) = \begin{cases} 1 & if \quad \Delta\varphi_{ij} \le 0 \\ exp\left(\dfrac{-\Delta\varphi_{ij}}{\Delta\bar{\varphi}T_k}\right) & if \quad \Delta\varphi_{ij} > 0 \end{cases} \tag{9}$$

where $\Delta\varphi_{ij} = \varphi(x_j) - \varphi(x_i)$, $\varphi(x_j)$ and $\varphi(x_i)$ are objective function values of candidate and current designs, $\Delta\bar{\varphi}$ is a normalization constant which is the running average of $\Delta\varphi_{ij}$, T_k is the strategy temperature. $\Delta\bar{\varphi}$ is updated as follows when $\Delta\varphi_{ij} > 0$ before computing the acceptance probability (Balling 1991):

$$\Delta\bar{\varphi} = \frac{M \times \Delta\bar{\varphi} + \Delta\varphi_{ij}}{M+1} \tag{10}$$

$$M = M + 1 \tag{11}$$

where M is the number of terms in the running average. The initial values for $\Delta\bar{\varphi}$ and M are taken as 1 and 0, respectively. The second half of the acceptance probability criterion is carried out by generating a random number rn, uniformly distributed over the interval [0,1]. The candidate design is accepted as the current design if $rn < A_{ij}$, otherwise the iterations continue with the previous design (Degertekin 2007).

The strategy temperature T_k is gradually decreased according to a cooling schedule. The acceptance probability of a neighbour design is higher at high temperature in order to prevent local optima. After the values of starting acceptance probability P_s and the final acceptance probability P_f and the number of temperature reduction cycles N are assigned, the starting temperature T_s, the final temperature T_f and the strategy temperature T_k are calculated as:

$$T_s = \frac{-1}{lnP_s} \tag{12}$$

$$T_f = \frac{-1}{lnP_f} \tag{13}$$

$$T_{k+1} = \alpha T_k \tag{14}$$

where α is the cooling factor and less than one. T_{k+1} is the temperature in the next cycle. After N cycles, the final temperature value T_f is expressed as

$$T_f = T_s \alpha^{N-1} \tag{15}$$

and α is obtained as

$$\alpha = \left(\frac{lnP_s}{lnP_f}\right)^{1/(N-1)} \tag{16}$$

As the acceptance probability of a candidate design is higher at high temperatures, the algorithm can escape from local minima easily and thermal equilibrium is reached in a fewer number of iterations. Acceptance probability is smaller at low temperatures, the algorithm needs larger number of iterations to escape from local minima and attain equilibrium. For that reason, it is necessary to increase the number of iterations while the temperature is reduced to reach the thermal equilibrium (Degertekin 2007). The number of iterations carried out during each temperature reduction cycle, *IPC(T)*, are given as (Bennage 1994, Bennage and Dhingra 1995):

$$IPC(T) = IPC_f + (IPC_f - IPC_s)\left(\frac{T - T_f}{T_f - T_s}\right) \tag{17}$$

where IPC_s is the number of iterations per cycle at the initial temperature T_s while IPC_f is the number of iterations per cycle needed at the final temperature T_f.

The steps described in Sections 5.2-5.4 are repeated until all design variables are selected only once. After all design variables are considered, an iteration is completed. Similarly, the same process is repeated as many as the number of iterations (*IPC*) of the current cycle. When it is completed, the next cycle starts with the current design. It should be also noted that if a neighbour design satisfies the aspiration criteria, acceptance probability is not applied to this design additionally.

5.5 Terminating criteria

The steps explained in Section 5.2-5.4 are repeated until a termination criterion is satisfied. In this study, two termination criteria were applied which are the same ones of the authors's previous work (Hayalioglu and Degertekin 2007). The first one stops the algorithm when a predetermined maximum cycle number is reached. The second criterion stops the process before the maximum cycle number, if a better design (lighter frame) than the current optimum is not encountered during a definite number of successive cycles in HTS.

6. Optimum design algorithm using a hybrid tabu-simulated annealing algorithm

The proposed HTS algorithm for optimum design of steel frames consists of the following steps:

Step 1: Initialize cycle counter $N = 0$, tabu list (*itl*) and aspiration list (*asp*) $itl = \emptyset$, $asp = \emptyset$. Assign the values of the maximum number of cycle (N_{max}), the definite number of successive cycles, neighbourhood depth (*mv*), P_s, P_f and calculate T_s, T_f and α using Eqs.(12), (13) and (16).

Step 2: Randomly generate 100 initial designs. Obtain the response of the frames. Calculate the values of $\varphi(x)$ using Eqs. (1)-(8). Select the one with the lowest $\varphi(x)$ value. Assign it as current design $x_c = x$, $\varphi(x_c) = \varphi(x)$. If x_c satisfies all the constraints, assign it also as current optimum design and record it in the aspiration list, $x_{asp}=x_c$, $\varphi(x_{asp}) = \varphi(x_c)$.

Step 3: Determine the iterations required per cycle *IPC* from Eq. (17). Initialize iteration counter $ic = 0$.

Step 4: Randomly select *i*-th design variable of x_c, $i \in [1,2,\ldots,ng]$ to be changed.

Step 5: Give a random perturbation (*irp*), $irp \in [-mv,\ldots,-1,1,\ldots,mv]$ to the variable *i* to generate a neighbour design (x_n) of x_c.

Step 6: Obtain response of x_n. Calculate the value of $\varphi(x_n)$ using Eqs.(1)-(8).

Step 7: Update tabu list. If x_n is tabu, go to step 8. Otherwise go to step 9.

Step 8: If x_n satisfies aspiration criteria (i.e. $\varphi(x_n) < \varphi(x_{asp})$), ignore the tabu status of x_n, go to step 12, else go to step 13.

Step 9: Calculate $\Delta\varphi(x_{cn}) = \varphi(x_n) - \varphi(x_c)$. If $\Delta\varphi(x_{cn}) \leq 0$, go to step 11, else go to step 10.

Step10: Update $\Delta\overline{\varphi}$ using Eq. (10). Calculate acceptance probability $A_{ij}(T_k)$ from the second half of Eq. (9). Generate an uniformly distributed random number *rn* over the interval [0,1]. If $rn < A_{ij}$, go to step 11. Otherwise go to step 13.

Step11: Accept x_n as the current design $x_c = x_n$, $\varphi(x_c) = \varphi(x_n)$. If x_n satisfies also aspiration criteria, update aspiration list $x_{asp} = x_n$, $\varphi(x_{asp}) = \varphi(x_n)$. Go to step 13.

Step12: Update aspiration list. Record x_n as the current design and the current optimum design as well. $x_c = x_n$, $\varphi(x_c) = \varphi(x_n)$, $x_{asp} = x_n$, $\varphi(x_{asp}) = \varphi(x_n)$. Go to step 13.

Step13: If all design variables are selected only once, go to step 14, else go to step 4.

Step14: If $ic \geq IPC$, go to step 15. Otherwise, set the iteration counter $ic = ic + 1$ and go to step 4.

Step15: Update the temperature T_k using Eq. (14). Set the cycle number $N = N + 1$. If the terminating criteria explained Section 5.5 are satisfied, terminate the algorithm and define the current optimum in the aspiration list as the final optimum. Otherwise go to step 3.

7. Benchmark examples

The structural analysis of steel frames and the proposed HTS algorithm were coded and executed in FORTRAN programming language. In this section, three steel frames were used to prove the effectiveness and reliability of HTS. They were previously optimized using GA, SA and TS (Hayalioglu and Degertekin

2007, Degertekin 2007, Degertekin *et al*. 2008). Therefore, material properties, design constraints and load combinations are taken the same as the ones of those articles. These values are shown in tabular form in Table 1.

The displacement constraints in Table 1 were increased by 30% to include the effect of the coefficient 1.3 in the AISC-LRFD (2001) wind load case. Two discrete section list comprised 64 W sections each were used in the design examples. The first one is beam section list taken from AISC-ASD (1989)-Part 2, "Beam and Girder Design"- Allowable stress design selection table for shapes used as beams. The boldface type sections (lighter ones) were selected starting from W36 × 720 to W12 × 19. The second one is column section list taken from the same code, Part 3, "Column Design"- Column W shapes tables. They were selected from W14 × 283 to W6 × 15. The maximum cycle number N_{max} and the cycle number for the second terminating criterion were selected as 200 and 30 respectively, in HTS.

In order to provide exact comparison with the benchmark examples, the same tuning parameter values were chosen for the execution of the HTS algorithm: P_s and P_f were selected 0.50 and 10^{-7}, respectively. The neighbourhood depth (mv) for the perturbations was selected as ±3. IPC_s and IPC_f were selected as 1 and 4 respectively. Five times the number of groups was assigned for the length of the tabu list. The penalty constant C was selected as 1.0 in HTS. A personal computer with the 3.2 GHz microprocessor was used for all computations. The effective length factor was calculated from the approximate equation proposed by Dumonteil (1992). The geometrically nonlinear analysis algorithm was performed for HTS algorithm is the same as the one of the previous work (Hayalioglu and Degertekin 2007). However, computational time will come an issue when larger, more complex frames are analysed. Any improvement to this step can be made by using linear analysis and then applying the moment magnification factors to consider the second order effects which is also a choice in lieu of the geometrically nonlinear analysis according to the AISC-LRFD (2001).

7.1 Design of three-storey, two-bay planar frame

The three-storey two-bay planar frame shown in Fig. 1 is the first benchmark example. The dimensions

Table 1 Design data for steel frames

Material properties	Modulus of elasticity, $E = 200$ GPa Shear modulus, $G = 83$ GPa Yield stress, $f_y = 248.2$ MPa Unit weight of material, $\rho = 7{,}850$ kg/m^3
Displacement constraints	Max. drift of the top storey = $H/500$-$H/400$ (H : total height of the frame) Interstorey drift = $h_c/300$ (h_c: the height of considered storey)
Strength constraints	AISC-LRFD (2001) interaction equations given in Eqs. (4)-(5)
Load combinations	I : $1.4D$ II : $1.2D + 1.6L + 0.5L_r$ III : $1.2D + 1.6L_r + 0.5L$ IV : $1.2D + 1.3W + 0.5L + 0.5L_r$ (D : dead load; L : live load; L_r : live roof load; W : wind load)

and the number of grouping are also shown in Fig. 1. This frame was designed using TS and SA (Hayalioglu and Degertekin 2007). The top and interstorey drift constraints were determined as 2.73 cm and 1.52 cm, respectively. The load values were arranged as: 32.85 kN/m for dead load (D), 21.9 kN/m for live load (L) and roof live load (L_r), 22.4 kN for wind load (W).

10 different designs were executed using HTS algorithm and the lightest one of those was reported in Table 2. Design history of the optimum frame weight for three-storey, two-bay frame was also depicted in Fig. 2.

HTS obtained the optimum design after 126 cycles with 24.63 min computing time. This shows that HTS could not find a lighter frame in 30 cycles after 96-th cycle. It is obvious that the design is controlled by strength constraints. The top storey and the maximum interstorey drift are far below from their boundary values as shown in Table 2.

HTS algorithm yielded lighter frame than SA and TS as reported in Table 2. Moreover, HTS required less computational effort than SA and TS.

7.2 Single-storey, 8 member space frame

The single-storey, 8 member space frame shown in Fig. 3 is the second example. This frame was designed using SA and GA in accordance with the AISC-ASD (1989) and AISC-LRFD (2001) specifications. SA yielded lighter frame than GA considering AISC-LRFD specification (Degertekin 2007).

The frame consists of three groups: 1st group: the beams in x-direction, 2nd group: the beams in y-direction, 3rd group: the all columns. The values of 3.12 kPa for dead load (D), 2.4 kPa for live load (L) and roof live load (L_r) were considered in this example. Wind loading was obtained from Uniform Building Code (1997) using the equation $p = C_e C_q q_s I_w$, where p is design wind pressure; C_e is combined

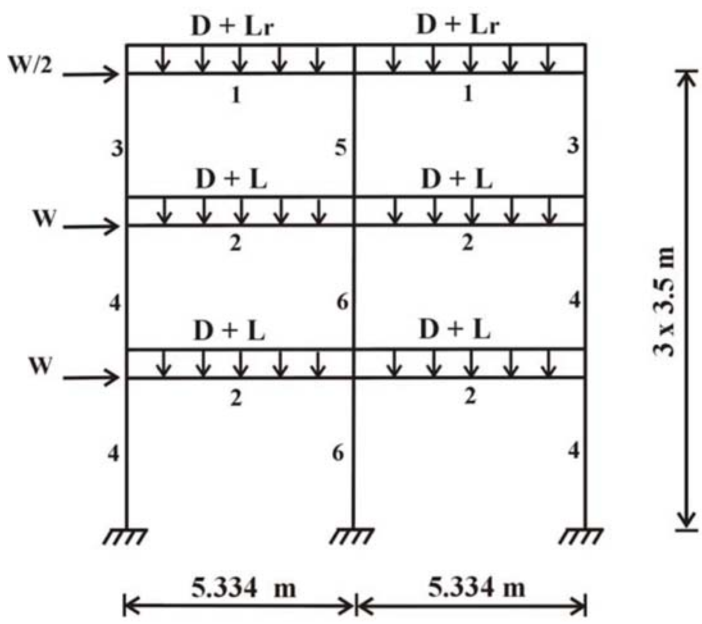

Fig. 1 Three-storey, two-bay planar frame

Table 2 Optimum design results of three-storey, two-bay planar frame

Group no.	HTS	TS	SA
	This study	Hayalioglu and Degertekin (2007)	
1	W 18 × 35	W 18 × 35	W 16 × 40
2	W 18 × 35	W 18 × 35	W 16 × 40
3	W 8 × 31	W 8 × 35	W 14 × 43
4	W 8 × 31	W 8 × 35	W 14 × 43
5	W 8 × 31	W 10 × 33	W 12 × 40
6	W 14 × 53	W 14 × 48	W 14 × 43
Weight (kg)	3355	3437	3910
Top storey drift (cm)	1.64	1.70	1.46
Max. interstorey drift (cm)	0.67	0.66	0.62
Max. interaction ratio	1.0	1.0	0.98
Computing time (min)	24.63	28.25	30.92

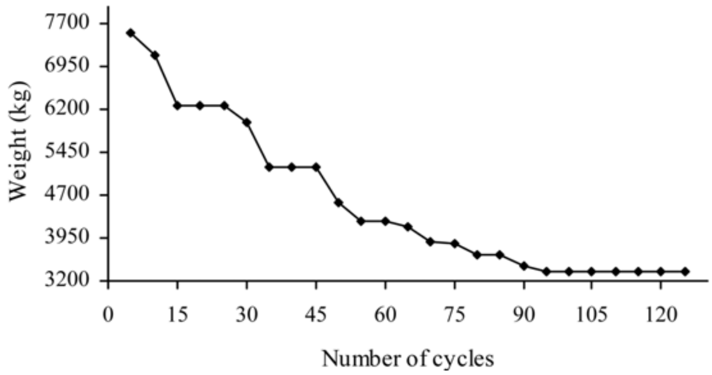

Fig. 2 Design history of the optimum frame weight for three-storey, two-bay planar frame

height, exposure and gust factor coefficient; C_q is pressure coefficient; q_s is wind stagnation pressure; and I_w is wind importance factor. Exposure D was assumed and the values for C_e were selected depending on the frame height and exposure type. The C_q values were assigned as 0.8 and 0.5 for inward and outward faces. The value of q_s was selected as 0.785 kPa assuming a basic wind speed of 129 km/h (80 mph) and the wind importance factor was assumed to be one. The horizontal loads due to wind act in the x-direction at each unrestrained node. The maximum top storey drift was restricted to 1.3 cm (Degertekin 2007).

For HTS algorithm, 10 independent frames were obtained generated from randomly selected 10 different initial designs and the lightest one of those was reported in Table 3. The convergence history of single-storey, 8-member space frame using HTS was also depicted in Fig. 4.

HTS algorithm outperformed in comparison with SA and GA algorithms. The optimum weight of 1,705 kg obtained using HTS after 118 cycles. It is 1.3% and 6.8% lighter than SA's and GA's. The drift constraints govern the design. HTS also consumed less computing time than SA, but more computing time than GA.

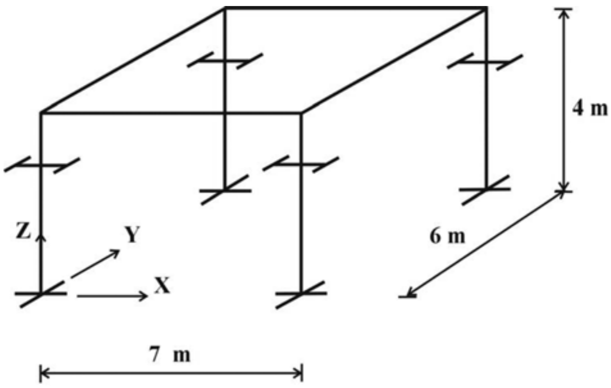

Fig. 3 Single-storey, 8-member space frame

7.3 Design of 4-storey, 84-member space frame

The last example is the 4-storey space frame with a plan and side view shown in Fig. 5. This frame was optimized using TS and GA (Degertekin *et al*. 2008).

The structure consists of 10 groups: 1st group: outer beams of 4th storey, 2nd group: outer beams of 3rd, 2nd and 1st storeys, 3rd group: inner beams of 4th storey, 4th group: inner beams of 3rd, 2nd and 1st storeys, 5th group: corner columns of 4th storey, 6th group: corner columns of 3rd, 2nd and 1st storeys, 7th group: outer columns of 4th storey, 8th group: outer columns of 3rd, 2nd and 1st storeys, 9th group: inner columns of 4th storey, 10th group: inner columns of 3rd, 2nd and 1st storeys. The wind loads act in the x-direction at each node on the sides AB and CD.

The top and interstorey drifts were restricted to 3.50 cm and 1.17 cm in the x-direction, respectively. The following load values were assigned as: 3.84 kPa for dead load, 2.4 kPa for live load and roof live load (Degertekin *et al*. 2008). The values of the wind loads were taken the same as the one of the previous example. 10 different frame designs were obtained and the lightest of them was shown in Table 4. Design history of single-storey, 8-member space frame using HTS was illustrated in Fig. 6.

HTS obtained the optimum frame with a weight of 17,913 kg after 174 cycles. It is 3.9% and 10.7%

Table 3 Optimum design results of single-storey, 8-member space frame

Group no.	HTS	SA	GA
	This study	Degertekin (2007)	
1	W 12×30	W 12×30	W 14×30
2	W 12×26	W 12×30	W 14×30
3	W 10×26	W 8×24	W 8×28
Weight (kg)	1705	1728	1830
Top storey drift (cm)	1.23	1.24	1.27
Max. interstorey drift (cm)	1.23	1.24	1.27
Number of analyses (cm)	3157	6120	2928
Max. interaction ratio	0.89	*	*
Computing time (min)	5.82	11.3	5.7

*Unavailable

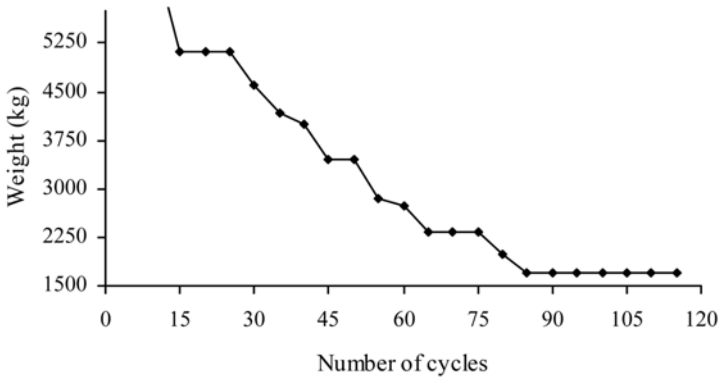

Fig. 4 Design history of the optimum frame weight for single-storey, 8-member space frame

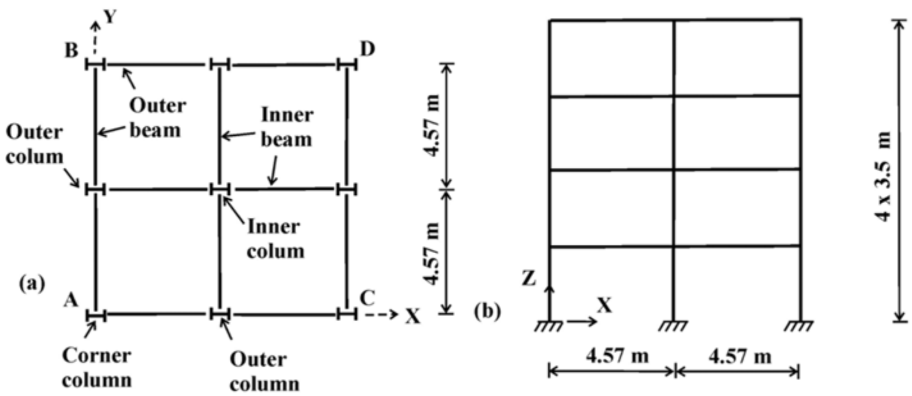

Fig. 5 Four-storey, 84-member space frame: (a) plan, (b) side view

lighter than the optimum frames obtained by TS and GA as given in Table 4. The drift constraints were active at the optimum and they controlled the design again. The number of structural analyses required by HTS was less than the TS's, but more than the GA's.

8. Conclusions

A hybrid tabu-simulated annealing heuristic algorithm is applied to the optimum design of steel frames for the first time. The benchmark examples presented in this study revealed that HTS is able to obtain lighter frames when compared to GA, TS and SA. The reason for this is that HTS uses the powerful tools of TS and SA simultaneously. It does not turn back to the old designs and therefore it is able to inspect different areas in the solution space. In addition to obtaining lighter frames, HTS required less computational effort than TS and SA. However; it should be noted that the computing time associated with HTS is still longer than GA. The authors think that computational effort could be reduced by using GA-based hybrid algorithms and the alternative nonlinear analysis procedure proposed by AISC-LRFD (2001). Hence, future work may focus on the hybrid GA-TS and GA-SA algorithms and the alternative nonlinear analysis procedure.

Table 4 Optimum design results of four-storey, 84-member space frame

Group no.	HTS	TS	GA
	This study	Degertekin *et al*. (2008)	
1	W 14×30	W21×44	W16×31
2	W 14×30	W14×30	W16×31
3	W 14×30	W14×30	W16×50
4	W 21×44	W14×30	W24×55
5	W 8×31	W12×45	W10×39
6	W 14×43	W14×43	W14×48
7	W 10×33	W14×43	W8×31
8	W 10×33	W14×43	W10×33
9	W 10×33	W10×33	W8×31
10	W 14×43	W14×33	W14×48
Weight (kg)	17913	18637	20060
Top storey drift (cm)	3.42	3.50	3.32
Max. interstorey drift (cm)	1.11	1.17	1.08
Max. interaction ratio	0.91	*	*
Number of analyses	17115	21360	9120
Computing time (min)	180.1	237.2	112.5

*Unavailable

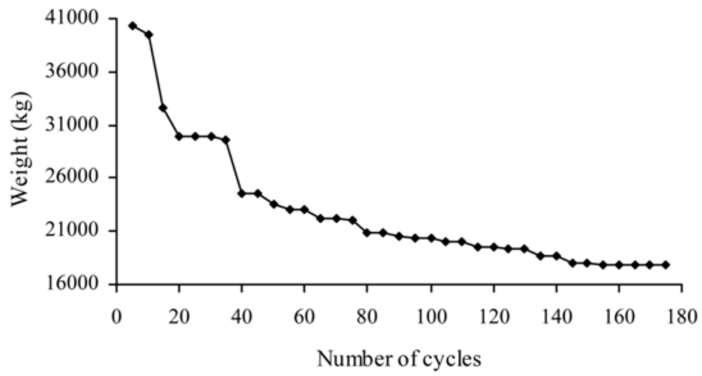

Fig. 6 Design history of the optimum frame weight for four-storey, 84-member space frame

References

American Institute of Steel Construction. (1989), *Manual of steel construction: allowable stress design,* Chicago, Illionis.

American Institute of Steel Construction. (2001), *Manual of steel construction: load and resistance factor design*, Chicago, Illionis.

Balling, R.J. (1991), "Optimal steel frame design by simulated annealing", *J. Struct. Eng. ASCE*, **117**(6), 1780-1795.

Bennage, W.A. (1994), "*Structural optimization using simulated annealing and tabu search*", MSc Thesis, The University of Wisconsin-Milwaukee.

Bennage, W.A. and Dhingra, A.K. (1995), “Single and multiobjective structural optimization in discrete continuous variables using simulated annealing”, *Int. J. Numer. Meth. Eng.*, **38**(16)**,** 2753- 2773.

Camp, C., Pezeshk, S. and Cao, G. (1998) “Optimized design of two-dimensional structures using a genetic algorithm”, *J. Struct. Eng. ASCE*, **124**(5), 551-559.

Chan, F.T.S., Chung, S.H. and Wadhwa, S. (2005) “A hybrid genetic algorithm for production and distribution”, *Omega- Int. J. Manage. S.*, **33**(4), 345-355.

Chan, F.T.S. and Kumar, V. (2009) “Hybrid TSSA algorithm-based approach to solve warehouse-scheduling problems”, *International Journal of Production Research,* DOI: 10.1080/00207540701305530.

Degertekin. S.O. (2007), “A comparison of simulated annealing and genetic algorithm for optimum design of nonlinear steel space frames”, *Struct. Multidiscip. O.*, **34**(4), 347-359.

Degertekin, S.O., Saka, M.P. and Hayalioglu, M.S. (2008), “Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm”, *Eng. Struct.*, **30**(1), 197-205.

Dhingra, A.K. and Bennage, W.A. (1995), “Discrete and continuous variable structural optimization using tabu search”, *Eng. Optimiz*., **24**(3), 177-196.

Dumonteil, P. (1992), “Simple equations for effective length factors”, *Eng. J. AISC*, **29**(3), 111-115.

Glover, F. (1989), “Tabu search-Part I”, *ORSA J. Comp.*, **1**(3), 190-206.

Glover, F. (1990), “Tabu search-Part II”, *ORSA J. Comp.*, **2**(1): 4-32.

Glover, F. and Laguna, M. (1997), *Tabu search*, Kluwer Academic Publishers, Norwell, Massachusetts.

Hayalioglu, M.S. (2000), “Optimum design of geometrically non-linear elastic-plastic steel frames via genetic algorithm”, *Comput. Struct.*, **77**(5): 527-538.

Hayalioglu, M.S. (2001), “Optimum load and resistance factor design of steel space frames using genetic algorithm”, *Struct. Multidiscip. O.*, **21**(4), 292-299.

Hayalioglu, M.S. and Degertekin, S.O. (2005), “Minimum cost design of steel frames with semi-rigid connections and column bases via genetic optimization”, *Comput. Struct.*, **83**(21-22), 1849-1863.

Hayalioglu, M.S. and Degertekin, S.O. (2007), “Minimum-weight design of non-linear steel frames using combinatorial optimization algorithms”, *Steel Compos. Struct.*, **7**(3), 201-217.

Holland, J.H. (1975), *Adaption in natural and artificial systems*, The Universtiy of Michigan Press, Ann Arbor, MI.

Huang, M.W. and Arora, J.S. (1997), “Optimal design steel structures using standard sections”, *Struct. Optim.*, **14**(1), 24-35.

Jenkins, W.M. (1992), “Plane frame optimum design environment based on genetic algorithm”, *J. Struct. Eng. ASCE*, **118**(11), 3103-3112.

Kameshki, E.S. and Saka, M.P. (2001), “Genetic algorithm based optimum bracing design of non-swaying tall plane frames”, *J. Constr. Steel Res.*, **57**(10), 1081-1097.

Kameski, E.S. and Saka, M.P. (2003), “Genetic algorithm based optimum design of nonlinear planar steel frames with various semirigid connections”, *J. Constr. Steel Res.*, **59**(1), 109-134.

Kargahi, M., Anderson, J.C. and Dessouky, M.M. (2006), “Structural weight optimization of frames using tabu search. I: Optimization procedure”, *J. Struct. Eng. ASCE*, **132**(12), 1858-1868.

Kargahi, M. and Anderson, J.C. (2006), “Structural weight optimization of frames using tabu search. II: Evaluation and seismic performance”, *J. Struct. Eng. ASCE*, **132**(12), 1869-1879.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), “Optimization by simulated annealing”, *Science*, **220** (4598), 671-680.

Liu, M., Burns, S.A. and Wen, Y.K. (2006), “Genetic algorithm based construction-conscious minimum weight design of seismic steel moment-resisting frames”, *J. Struct. Eng. ASCE*, **132**(1), 50-58.

Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. (1997), “A new simulated annealing-based tabu search algorithm for unit commitment”, *Proc. IEEE Int. Conf . Sys. Man. Cybernetics*, **3**, 2432-2437.

Metropolis, N., Rosenbluth, A,, Teller, A. and Teller E. (1953), “Equation of state calculations by fast computing machines”, *J. Chem. Phys.,* **21**, 1087-1092.

Mishra, N., Prakash, Tiwari, M.K, Shankar, R. and Chan, F.T.S. (2005), “Hybrid tabu-simulated annealing based approach to solve multi-constraint product mix decision problem”, *Expert Syst. Appl.*, **29**(2), 446-454.

Ongsakul, W. and Bhasaputra, P. (2002), “Optimal power flow with FACTS devices by hybrid TS/SA approach”, *Int. J. Elect. Power*, **24**(10), 851-857.

Park, H.S. and Sung, C.W. (2002), “Optimization of steel structures using distributed simulated annealing

algorithm on a cluster of personal computers", *Comput. Struct.*, **80**(14-15), 1305-1316.
Pezeshk, S., Camp, C.V. and Chen, D. (2000), "Design of nonlinear framed structures using genetic optimization", *J. Struct. Eng. ASCE*, **126**(3), 382-388.
Rao, A.R.M. and Arvind, N. (2007), "Optimal stacking sequence design of laminated composite structures using tabu embedded simulated annealing", *Struct. Eng. Mech.*, **25**(2), 239-268.
Swarnkar, R. and Tiwari, M.K. (2004), "Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based approach", *Robot. CIM-Int. Manuf.*, **20**(3), 199-209.
Toropov, V.V. and Mahfouz, S.Y. (2001), "Design optimization of structural steelwork using a genetic algorithm, FEM and a system of design rules", *Eng. Computation*, **18**(3-4), 437-459.
Uniform Building Code. (1997), *International Conference of Building Officials*, Whittier, California.
Zhang, D.F. and Deng, A.S. (2005), "An effective hybrid algorithm for the problem of packing circles into a larger containing circle", *Comput. Oper. Res.*, **32**(8), 1941-1951.
Zolfaghari, S., Liang, M. (1999), "Jointly solving the group scheduling and machining speed selection problems: a hybrid tabu search and simulated annealing approach", *Int. J. Production Res.*, **37**(10), 2377-2397.

CC