Steel and Composite Structures, Vol. 7, No. 3 (2007) 201-217
DOI: http://dx.doi.org/10.12989/scs.2007.7.3.201

Minimum-weight design of non-linear steel frames using combinatorial optimization algorithms

M. S. Hayalioglu† and S. O. Degertekin‡

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

(Received December 11, 2006, Accepted March 8, 2007)

Abstract. Two combinatorial optimization algorithms, tabu search and simulated annealing, are presented for the minimum-weight design of geometrically non-linear steel plane frames. The design algorithms obtain minimum weight frames by selecting suitable sections from a standard set of steel sections such as American Institute of Steel Construction (AISC) wide-flange (W) shapes. Stress constraints of AISC Load and Resistance Factor Design (LRFD) specification, maximum and interstorey drift constraints and size constraints for columns were imposed on frames. The stress constraints of AISC Allowable Stress Design (ASD) were also mounted in the two algorithms. The comparisons between AISC-LRFD and AISC-ASD specifications were also made while tabu search and simulated annealing were used separately. The algorithms were applied to the optimum design of three frame structures. The designs obtained using tabu search were compared to those where simulated annealing was considered. The comparisons showed that the tabu search algorithm yielded better designs with AISC-LRFD code specification.

Keywords: optimum design; tabu search; simulated annealing; steel frames; non-linear analysis; allowable stress design; load and resistance factor design.

1. Introduction

In recent years, some local search algorithms such as tabu search (TS) and simulated annealing (SA) have been used to solve many optimization problems. Local search is an emerging paradigm for combinatorial search which has recently been shown to be very effective for a large number of combinatorial problems. It is based on the idea of navigating the search space by iteratively stepping from one solution to one of its neighbours, which are obtained by applying a simple local change to it.

TS is based on the human memory process and uses an iterative neighbourhood search procedure in an attempt to avoid becoming trapped in local optima. TS was developed by Glover (1989, 1990) for solving combinatorial optimization problems. It has been applied to many different fields of engineering and technology recently (Chamberland and Sanso 2002, Sait and Zahra 2002, Abido 2002, Richards and Gunn 2003, Pai *et al*. 2003, Cunha and Riberio 2004, Jeon and Kim 2004). An excellent theoretical aspect of TS is given by Glover and Laguna (1997).

TS is also suitable for discrete design variables. It was found from literature survey that the applications of TS to the structural optimization were only about the optimal design of planar and space

†Ph.D., Professor, Corresponding Author, E-mail: hsedat@dicle.edu.tr
‡Ph.D., Research Assistant

trusses which behave linear elastically (Bennage 1994, Dhingra and Bennage 1995a, Bland 1995, Bennage and Dhingra 1995b, Bland 1998a, Bland 1998b, Manoharan and Shanmuganathan 1999).

SA is an application of annealing process in solids into the computational algorithms and able to solve discrete and continuous optimization problems. SA method was originally put forward by Kirkpatrick *et al*. (1983) for optimization problems. SA has been used in different fields of engineering optimization problems recently (Teegavarapu *et al*. 2002, Erdal and Sonmez 2005, Zhao and Zeng 2006, Sathiya *et al*. 2006, Kumar *et al*. 2006, Moita *et al*. 2006).

SA also deals with the discrete and continuous optimum structural design problems. SA was applied to the size and/or topological optimization of metal structures, ie plane frames, plane and/or space trusses subjected to static or dynamic loading using discrete and/or continuous variables (Bennage and Dhingra 1995a, Dhingra and Bennage 1995b, Pantelidis and Tzan 2000, Chen and Su 2002, Hasancebi and Erbatur 2002, Park and Sung 2002). SA was also applied to the optimum design of concrete retaining structures (Ceranic *et al*. 2001) and shell structures (Barski 2006). Rama Mohan Rao and Arvind (2007) put forward a SA algorithm in which TS is embedded in the algorithm in order to prevent recycling of recently visited solutions. They used this algorithm for optimal stacking sequence design of laminate composite structures.

As regards the using of SA in the optimum design of steel frames under the actual design constraints and loads of code specifications, the following articles can be considered: Huang and Arora (1997) employed SA in the optimum design of steel plane frames subjected to design constraints of AISC-ASD specification. They used standard AISC-W sections as discrete variables and continuous variables establishing relationships between the cross-sectional properties and applying the rounding-off procedure. Balling (1991) applied SA to the optimum design of 3-D steel frames for the design constraints of AISC-ASD specification using discrete W steel sections and compared the results with the ones of branch and bound method. The behaviour of the structures in the last two articles was assumed to be linear-elastic.

The present article aims at applying/comparing two combinatorial optimization algorithms, tabu search and simulated annealing, for the optimal design of steel plane frames with different design code and behaviour from the ones of aforementioned articles, ie geometrically non-linear steel frames, under the actual design constraints of code specifications (AISC-LRFD 2001). Displacement constraints and size constraints due to constructional reasons were also incorporated into the optimal design of frames. Discrete design variables selected from the standard set of AISC wide-flange (W) shapes were used. The article also compares two design codes, AISC-LRFD (2001) and AISC-ASD (1989) specifications, for the optimal design of steel frames. Optimum designs of three steel frames were performed using TS and SA methods together with LRFD and ASD codes. The two methods were compared to each other according to the results of the numerical design examples. The two design codes were also compared while each of the two methods was used.

2. The formulations of the optimum design problem

The discrete optimum design problem of non-linear steel frames where the minimum weight is considered as the objective can be stated as follows:

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

subjected to the stress constraints of AISC-LRFD (2001) and AISC-ASD (1989), displacement and size constraints. In Eq. (1), *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.

All the constraints are given in normalized forms which are suitable for TS and SA whose objective functions can be arranged in an unconstrained manner.

The displacement constraints are

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

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

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 stress constraints taken from AISC-LRFD (2001) are expressed in the following equations.

For members subject to bending moment and axial force:

for $$\frac{P_u}{\phi P_n} \ge 0.2$$

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

for $$\frac{P_u}{\phi P_n} \ge 0.2$$

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

where *nm* is total number of members in the frame.

If the axial force in a member is tension, the terms in Eqs. (4) and (5) can be defined as: P_u = required tensile strength, P_n = nominal tensile strength, M_{ux} = required flexural strength about the major axis including second-order effects (geometric non-linearity), M_{nx} = nominal flexural strength about the major axis, $\phi = \phi_t$ = resistance factor for tension (equal to 0.90), ϕ_b = resistance factor for flexure (equal to 0.90).

If the axial force in a member is compression, the terms in Eqs.(4) and (5) are described as: P_u = required compressive strength, P_n = nominal compressive strength, $\phi = \phi_c$ = resistance factor for compression (equal to 0.85). The definition of the other terms is the same as the previous one. The nominal compressive strength of a member is computed as

$$P_n = A_g \,.\, F_{cr} \tag{6}$$

$$F_{cr} = (0.658^{\lambda_c^2})F_y \quad \text{for} \quad \lambda_c \le 1.5 \tag{7}$$

$$F_{cr} = \left(\frac{0.877}{\lambda_c^2}\right)F_y \qquad \text{for} \quad \lambda_c > 1.5 \tag{8}$$

$$\lambda_c = \frac{KL}{r\pi}\sqrt{\frac{F_y}{E}} \tag{9}$$

where A_g is cross-sectional area of a member; K is effective length factor; E is modulus of elasticity; r is governing radius of gyration; L is the member length; and F_y is the yield stress of steel. The effective length factor K, for unbraced frames were calculated from the following approximate equation (Dumonteil 1992):

$$K = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \tag{10}$$

where G_A and G_B are relative stiffness factors at A-th and B-th ends of column. The out-of-plane effective length factor for each column member was assumed to be one. The required strengths were determined from the non-linear analysis of steel frames subjected to the factored load combinations.

When the stress constraints of AISC-ASD (1989) are used the following equations are required to be included in the design formulation.

For members subjected to both axial compression and bending stresses,

$$g_i(x) = \left[\frac{f_a}{F_a} + \frac{C_{mx} f_{bx}}{\left(1 - \frac{f_a}{F'_{ex}}\right)F_{bx}}\right]_i - 1.0 \le 0\,, \qquad i = 1,\ldots,\, nc \tag{11}$$

$$g_i(x) = \left[\frac{f_a}{0.60F_y} + \frac{f_{bx}}{F_{bx}}\right]_i - 1.0 \le 0\,, \qquad i = 1,\ldots,\, nc \tag{12}$$

When $f_a / F_a \le 0.15$, Eq. (13) is permitted in lieu of Eqs. (11) and (12)

$$g_i(x) = \left[\frac{f_a}{F_a} + \frac{f_{bx}}{F_{bx}}\right]_i - 1.0 \le 0\,, \qquad i = 1,\ldots,\, nc \tag{13}$$

For members subjected to both axial tension and bending stresses,

$$g_i(x) = \left[\frac{f_a}{F_t} + \frac{f_{bx}}{F_{bx}}\right]_i - 1.0 \le 0\,, \qquad i = 1,\ldots,nb \tag{14}$$

where nc is total number of members subjected to both axial compression and bending stresses and nb is total number of members subjected to both axial tension and bending stresses.

In Eqs. (11)-(14), the subscript x, combined with subscripts b, m and e, indicate the axis of bending about which a particular stress or design property applies, and F_a = axial compressive stress that would be permitted if axial force alone existed, F_b = compressive bending stress that would be permitted if bending moment alone existed, F'_e = Euler stress divided by a factor of safety, f_a =

computed axial stress, f_b = computed compressive bending stress at the point under consideration, C_m = a coefficient whose value is taken as 0.85 for compression members in frames subject to sidesway. In Eq. (14), f_b is the computed bending tensile stress, f_a is the computed axial tensile stress, F_b is the allowable bending stress and F_t is the governing allowable tensile stress. Allowable, $0.6F_y$ and Euler stresses are increased by 1/3 in accordance with the specification when produced by wind, acting alone or in combination with the design dead and live loads. Definitions of the allowable and Euler stresses and the other details are given AISC-ASD (1989) specifications and therefore will not be repeated here.

The same normalized displacement constraints are used for AISC-ASD code as in Eqs. (2) and (3) with the exception that the subscript l is omitted due to single load condition.

The size constraints employed for constructional reasons given as follows:

$$g_n(x) = \frac{d_{un}}{d_{bn}} - 1.0 \le 0\,, \qquad n = 1,..,ncl \tag{15}$$

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 $\varphi(x)$ is then written for AISC-LRFD code as

$$\varphi(x) = W(x)\left[1 + C\left(\sum_{j=1}^{m}\sum_{l=1}^{nl} v_{jl} + \sum_{j=1}^{ns}\sum_{i=1}^{nsc}\sum_{l=1}^{nl} v_{jil} + \sum_{i=1}^{nm}\sum_{l=1}^{nl} v_{il} + \sum_{n=1}^{ncl} v_n\right)\right] \tag{16}$$

where C is a penalty constant to be selected depending on the problem. v_{jl}, v_{jil}, v_{il} and v_n are violation coefficients which are calculated as

$$\begin{array}{lll}
\text{if } g_{jl}(x) > 0 & \text{then} & v_{jl} = g_{jl}(x) \\
\text{if } g_{jl}(x) \le 0 & \text{then} & v_{jl} = 0 \\
\text{if } g_{jil}(x) > 0 & \text{then} & v_{jil} = g_{jil}(x) \\
\text{if } g_{jil}(x) \le 0 & \text{then} & v_{jil} = 0 \\
\text{if } g_{il}(x) > 0 & \text{then} & v_{il} = g_{il}(x) \\
\text{if } g_{il}(x) \le 0 & \text{then} & v_{il} = 0 \\
\text{if } g_{n}(x) > 0 & \text{then} & v_{n} = g_{n}(x) \\
\text{if } g_{n}(x) \le 0 & \text{then} & v_{n} = 0
\end{array} \tag{17}$$

For AISC-ASD code, the subscript l is omitted in Eqs. (16) and (17) together with the summation symbol associated with l, and the stress constraints in Eqs. (11)-(14) are used instead of Eqs. (4) and (5).

The minimum of the unconstrained function $\varphi(x)$ will be searched by TS and SA. It is clear that computation of $\varphi(x)$ for TS and SA requires the values of displacements and stresses in the frame. This is achieved by carrying out the non-linear analysis of frames.

3. Non-linear analysis of steel frames

In this study, an algorithm and its programming code were used developed by Levy and Spillers (1994) for the analysis of geometrically non-linear frames. A frame becomes geometrically non-linear mainly due to the presence of large deformations. The ultimate load is reached by incremental loads and at each load increment, incremental displacements are obtained by solving the equilibrium equations of the frame systems which are written in the incremental form of load and displacements. The member stiffness matrices are composed of linear elastic and geometric stiffness matrices. The iterations at each load increment continue until the unbalanced joint forces become quite small. The equilibrium equations are linearized and written for the deformed frame systems at each iteration. The solutions for all load increments are added up to acquire a total non-linear response.

4. Tabu search

TS is an optimization method which finds optimum solution by neighbourhood search in the solution space. A constrained optimization problem consists of constraints to be satisfied and an objective function whose minimum value is searched. Objective function is composed of design variables. Design variables are selected from a list of discrete variables that each of them is represented by a sequence number in that list.

First an initial design is generated randomly. A variable of this design is also selected randomly and various designs are obtained by changing only that variable in the range of a predetermined neighbourhood depth. For example, if the neighbourhood depth is determined as ±2, four different designs are obtained by exchanging the selected variable with two upper and lower variables in the sequence of the list. The best of the four designs is found (the best design is the one with the lowest objective function value). Meanwhile, the move (design variable) which determines the best design is recorded in a one-dimensional list called as ‘tabu list’. The other design variables of the best design are also checked whether they are in the tabu list or not. This design is replaced with the current design even if a design variable of it is not in the tabu list and the process continues starting with the new current design. The other design variables are also selected randomly and the same process is applied to each of them. A cycle is completed when all design variables are considered. The best of the neighbourhood designs is recorded in a list with single member if it satisfies all the constraints. This list is called ‘aspiration list’. The aspiration list is updated throughout the cycles when a better feasible design is encountered. During the search process, even if all variables of a best neighbourhood design are in the tabu list, its tabu status is temporarily ignored providing that it is a better feasible design than the one in the aspiration list. These two conditions are called ‘aspiration criteria’. This design is accepted as new current design and also put into the aspiration list. This design is rejected when it does not satisfy the aspiration criteria.

Tabu list is short term memory feature, which is continually refreshed as the search explores the design space. It 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.

5. Simulated annealing

The SA algorithm is inspired by the analogy between the annealing of solids and searching the solutions to optimization problems. Annealing is a thermal process applied to solids heating up them to a maximum temperature value at which all molecules of the solid crystal randomly arrange themselves in the liquid phase. The temperature of the molten crystal is fallen slowly later on. The crystalline structure becomes very tidy if the maximum temperature is quite high and the cooling is performed slowly enough. In this case, all molecules arrange themselves in the minimum energy (ground state). The solid reaches thermal equilibrium at each temperature level T described by a probability of being in a state i with energy E_i given by the Boltzman distribution:

$$P_r\{E = E_i\} = \frac{1}{Z(T)}\exp\left(\frac{-E_i}{k_B T}\right) \tag{18}$$

where $Z(T)$ is a normalization factor and k_B is Boltzmann constant. The Boltzmann distribution focuses on the states with lowest energy as the temperature decreases.

An analogy between the annealing and the optimization can be established in the following way: the energy of the solid denotes the objective (cost) function while the different states during the cooling represent the different solutions (designs) throughout the optimization process and ground state (minimum energy) denotes the global optimum.

In this study, a SA algorithm taken from Bennage and Dhingra (1995a) was employed but only the normalization constant in the Metropolis algorithm (Metropolis *et al.* 1953) which will be explained in advance was used as suggested by Balling (1991).

A constrained optimization problem with discrete design variables was also considered in SA like TS. First an initial design is generated randomly. The value of the unconstrained objective function of this design is obtained as explained in Section 2. The initial design is assigned as current design. A variable of this design is also selected randomly. This variable is exchanged with a randomly selected variable in the range of a predetermined neighbourhood depth $\pm mv$ to obtain a new design. For example, if the neighbourhood depth is determined as ±3 and the number 2 is selected in the interval [-3,+3] randomly, the above mentioned variable is exchanged with the one in the second upper row from that in the sequence of the list. (If the number were chosen as -3, the variable would be replaced with the one in the third lower row). This new design is called candidate design or neighbourhood design.

The value of the unconstrained objective function, which will be merely called objective function, of the candidate design is also calculated. The current design is replaced with the candidate design providing that the objective function value of the candidate design is less than the one of the current design, otherwise the candidate design is accepted or rejected according to the Metropolis algorithm: The acceptance probability A_{ij} of accepting candidate design j which has been generated from current design i is given as

$$A_{ij}(T_k) = \begin{cases} 1 \ \text{ if } \ \Delta\varphi_{ij} \le 0 \\ \exp\left(\dfrac{-\Delta\varphi_{ij}}{\Delta\bar{\varphi}T_k}\right) \ \text{ if } \ \Delta\varphi_{ij} > 0 \end{cases} \tag{19}$$

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):

$$\Delta\bar{\varphi} = \frac{M \times \Delta\bar{\varphi} + \Delta\varphi_{ij}}{M+1} \tag{20}$$

$$M = M + 1 \tag{21}$$

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.

The strategy temperature T_k is gradually decreased while the annealing process proceeds according to a cooling schedule. This requires the values for T_s and T_f, the initial and final values of temperature. Moreover, the values of starting acceptance probability P_s for $\Delta\varphi_{ij} = \Delta\bar{\varphi}$, the final acceptance probability P_f for $\Delta\varphi_{ij} = \Delta\bar{\varphi}$, and the number of temperature reduction cycles N are needed. Once P_s is given, the starting temperature T_s is found as

$$P_s = \exp\left(\frac{-1}{T_s}\right) \tag{22}$$

$$T_s = \left(\frac{-1}{lnP_s}\right) \tag{23}$$

The strategy temperature T_k is reduced as

$$T_{k+1} = \alpha T_k \tag{24}$$

where α is the cooling factor and less than one. T_{k+1} is the temperature in the next cycle. While T approaches zero, for design transitions with $\Delta\varphi_{ij} > 0$, A_{ij} also approaches zero. Given P_f, T_f is also obtained as

$$P_f = \exp\left(\frac{-1}{T_f}\right) \tag{25}$$

$$T_f = \left(\frac{-1}{lnP_f}\right) \tag{26}$$

After N cycles, the final temperature value T_f can be expressed as

$$T_f = T_s\alpha^{N-1} \tag{27}$$

and α is obtained as

$$\alpha = \left(\frac{lnP_s}{lnP_f}\right)^{1/(N-1)} \tag{28}$$

In an optimization problem with n design variables, an iteration around current design i comprises n designs generated by randomly perturbing one of the n variables providing that each variable is changed only once and temperature is kept constant during these neighbourhood search. The variables are selected randomly for perturbation.

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 rapidly, ie in a fewer number of iterations. As the aforementioned 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. The number iterations carried out during each temperature reduction cycle, $IPC(T)$, are given as

$$IPC(T) = IPC_f + (IPC_f - IPC_s)\left(\frac{T - T_f}{T_f - T_s}\right) \tag{29}$$

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. $IPC_s = 1$ and $IPC_f \in [3,6]$ work quite well for most structural optimization problems (Bennage and Dhingra 1995a).

In the present work, two terminating criteria were adopted which are the same for TS and SA. The first one stops the optimization process when a predetermined total number of cycles (the maximum cycle number) are performed. The second criterion stops the process before it reaches the maximum cycle number, providing a lighter frame is not found during a definite number of cycles.

6. Optimum design algorithm for tabu search

Design variables which form the objective function are discrete ones and they are the members of the steel section list with selected length. The algorithm will be given according to the stress constraints of AISC-LRFD code and it is the same algorithm for AISC-ASD code providing that the stress constraints of AISC-ASD and the displacement constraints for single loading condition are used as mentioned in Section 2. The optimum design algorithm for non-linear steel space frames using TS consists of the following steps:

1. Construct the initial design randomly. Assign this design as current design. Carry out the non-linear analysis and obtain the response of the frame. Calculate the value of unconstrained objective function $\varphi(x)$ using Eqs. (1)-(10) and (15)-(17).
2. Randomly select a variable (a member group of frame) of this design and obtain new designs by changing the variable along the neighbourhood depth. Carry out non-linear analysis for each new design and select the one with lowest objective function value (the best design) even though it does not satisfy all the constraints. Record the move (sequence number of the member group in the steel section list) which determines the best design in the tabu list. Put the best design into the aspiration list if it satisfies all the constraints.
3. Select again randomly a variable among the remaining ones of the best design and apply the process in the previous step to this design. Put the best neighbourhood design into the aspiration list if it satisfies the aspiration criteria, otherwise reject it.
4. Assign the best neighbourhood design as the current design and repeat steps 2 to 4 until the same process is completed for the last variable. This is the end of a cycle.
5. Start the next cycle with the current design obtained at the end of the previous cycle. Repeat steps 2 to 5 until one of the terminating criteria explained in Section 5 is satisfied. Define the existing design in the aspiration list at the end of the last cycle as the optimum design.

7. Optimum design algorithm for simulated annealing

The optimum design algorithm for non-linear steel frames using SA consists of the following steps:

1. Assign the values of P_s, P_f and N and calculate the cooling schedule parameters T_s, T_f and α from Eqs. (23), (26) and (28). Initialize the cycle counter $ic = 1$. Set the variable counter iv, and iterations per cycle counter il to zero.
2. Construct the initial design x_0 randomly. Assign this design as current design x_i. Carry out the non-linear analysis and obtain the response of the frame. Calculate the value of the objective function $\varphi(x_0)$ using Eqs. (1)-(10) and (15)-(17).
3. Determine the iterations required per cycle IPC from Eq. (29).
4. Randomly select a variable (the section no. of frame's member group) $ka \in [1,\ldots,n]$ to be changed and set $iv = iv + 1$. Give a random perturbation to the variable ka to generate a candidate design x_j in the neighbourhood of current design x_i.
5. Calculate the related objective function value $\varphi(x_j)$ and $\Delta\varphi_{ij} = \varphi(x_j) - \varphi(x_i)$.
6. If $\Delta\varphi_{ij} \leq 0$, accept the candidate design x_j as the current design. Meanwhile, if the new current design is a feasible one and better than the previous optimum, assign it temporarily as the optimum design. If $iv > n$ go to step 9, else return to step 4.
7. If $\Delta\varphi_{ij} > 0$, update $\Delta\bar{\varphi}$ using Eq. (20). Calculate the acceptance probability $A_{ij}(T_k)$ from the second half of Eq. (19). Generate a uniformly distributed random number rn over the interval [0,1]. If $rn<A_{ij}$ go to step 8, otherwise check if $iv > n$. If it is so, go to step 9, else go to step 4.
8. Accept the candidate design as the current design, ie set $x_i = x_j$ and $\varphi(x_i) = \varphi(x_j)$. If the new current design is a feasible one and better than the previous optimum, assign it temporarily as the optimum design. If $iv>n$ go to step 9, else go to step 4.
9. If $il \leq IPC$, set $iv = 0$, $il = il + 1$, and go to step 4. Otherwise go to step 10.
10. Update the temperature T_k using Eq. (24), set the cycle counter $ic = ic + 1$. If one of the terminating criteria explained in Section 5 is satisfied, terminate the algorithm and define the last temporary optimum as the final optimum design. Otherwise set $il = iv = 0$ and return to step 3.

8. Design examples

The algorithms were applied to the optimum designs of three frame structures. A36 steel grade with a modulus of elasticity of 200 GPa and shear modulus of 83 GPa was used in the examples. The yield stress and unit weight of material are 248.2 MPa and 7850 kg/m^3, respectively. Four different types of loads are employed: dead load (D), live load (L), roof live load (L_r), and wind loads (W). Four load combinations were taken into account per AISC-LRFD specification: I: ($1.4D$), II: ($1.2D + 1.6L + 0.5L_r$), III: ($1.2D + 1.6L_r + 0.5L$), and IV: ($1.2D + 1.3W + 0.5L + 0.5L_r$).However, only a load combination was considered per AISC-ASD specification: ($D + L + L_r + W$).

The values of 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) were considered in the three design examples. For the designs based on AISC-ASD, the maximum drift of top storey was restricted to H/500 (Ad Hoc Committee 1986), where H is the total height of the structure; the interstorey drift was also limited to h_c/300 (Ad Hoc Committee 1986), where h_c is the height of the considered storey. These limits were increased by 30% to include the effect of the coefficient 1.3 in the LRFD wind load combination.

Two discrete design sets for both algorithms and design codes 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 aim of the using the size constraint for columns as well as two separate section lists for columns and beams is to ensure physically realizable optimal designs.

The maximum cycle number N and the cycle number for the second terminating criterion were selected as 200 and 30 respectively, in TS and SA.

8.1 Design of one-storey, single-bay frame

The dimensions and member grouping of one-storey single-bay frame are shown in Fig. 1. The maximum top storey drift was restricted to 0.8 cm and 1.04 cm for ASD and LRFD respectively.

The following parameter values were chosen for the execution of the SA algorithm: The annealing algorithm is quite sensitive to the values of the starting acceptance probability (P_s) and the final acceptance probability (P_f). P_s and P_f were selected 0.50 and 10^{-7} respectively. Using higher values for P_s and P_f caused non-optimal solutions, while lower values for P_s resulted in premature convergence. Another parameter affecting the results is neighbourhood depth. The neighbourhood depth (*mv*) for the perturbations was selected as ±3. SA became inefficient when lower values for *mv* were used. Higher values for *mv* yielded local optima. IPC_s and IPC_f were selected as 1 and 4 respectively. The higher values of them did not improve the optimum design, while the lower values of IPC_f caused the non-optimal solutions.

The following parameter values were used for the TS algorithm: TS was quite sensitive for values of the neighbourhood depth. It was selected as ±3 in TS. The higher values of *mv* did not improve the designs to large extent and increased the computing time considerably, while the lower values of it caused local optima. It is quite important to assign an adequate value to the length of the tabu list in TS. The length should be neither so short nor so long. The short tabu list caused the searched to turn around the old designs while the long list restricted the search to a small area because most of the moves were in the tabu list. As a result of computational experience, five times the number of groups was found suitable for the length of the tabu list.

The penalty constant C was found to work best for the value of 1.0 in both algorithms. The lower

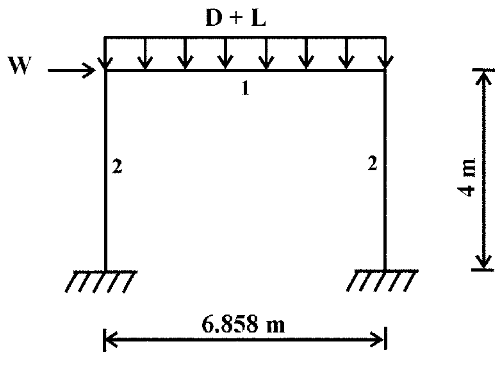

Fig. 1 One-storey, single-bay frame

Table 1 Optimum design results of one-storey, single-bay frame

Group No.	Tabu Search		Simulated Annealing	
	LRFD	ASD	LRFD	ASD
1	W 21×62	W 24×76	W 24×68	W 24×84
2	W 10×33	W 10×33	W 8×31	W 8×35
Weight (kg)	1029	1171	1068	1275
Top storey drift (cm)	0.73	0.56	1.04	0.77
Max. interstorey drift (cm)	0.73	0.56	1.04	0.77
Max. interaction ratio	0.99	1.0	0.93	0.91
Computing time (min)	2.24	0.50	2.52	0.53

values for C led to local optima, while higher values of it caused premature convergence. The second terminating criterion stops the optimization process if a better design is not found during 30 cycles. The values of cycles lower than 30 led to non-optimal solutions, whereas the values higher than 30 did not improve the optimum designs.

For each of the four design algorithms, 10 different optimum frames were obtained generated from randomly selected 10 different initial designs and the lightest one of those were reported in Table 1. A personal computer with the Pentium 4 - 3.2 GHz microprocessor was used for all computations.

The optimum designs were obtained after 63 and 59 cycles for LRFD and ASD respectively using TS, that is, the algorithm could not find a better design during the 30 cycles after the 33-rd and 29-th cycles. When SA was used, the optimum designs were obtained after 117-th cycles and 129-th cycles for LRFD and ASD, respectively. This implies that SA could not find a better design during the 30 cycles after 87-th and 99-th cycles for LRFD and ASD.

Stress constraints were active, while top and interstorey drift constraints were passive for both design codes at the optimum while TS was performed. Displacement and stress constraints were active for LRFD and ASD respectively, when SA was executed. TS yielded 3.7% lighter frame than the one obtained by SA for the LRFD code. TS resulted in 8.2% lighter frame than the one obtained by SA for the ASD code.

When compared the design codes, LRFD resulted in 12.1% and 16.2% lighter frames than the ones of ASD for the TS and SA implementations, respectively. Moreover; the computing time spent for TS was smaller than the one of SA as shown in the Table 1.

8.2 Design of 3-storey, 2-bay frame

Fig. 2 shows the frame configuration, dimensions and member grouping of the three-storey, two-bay frame. The top and interstorey drift constraints were determined as 2.1 cm and 1.17 cm respectively, for ASD. For each of the four design algorithms, 10 different designs were executed and the lightest one of those were reported in Table 2.

The optimum designs were obtained after 84 and 72 cycles for LRFD and ASD respectively, using TS algorithm. SA obtained the optimum designs at the end of the 144-th and 137-th cycles for LRFD and ASD, respectively. The optimum design results are given in Table 2. Stress constraints were active, top and interstorey drift constraints were passive in the four design results.

In the comparison of the algorithms, it was found that TS resulted in 12.1% lighter frame than the one of SA for LRFD code. %5.6 lighter frame was also obtained with TS than the one obtained using SA

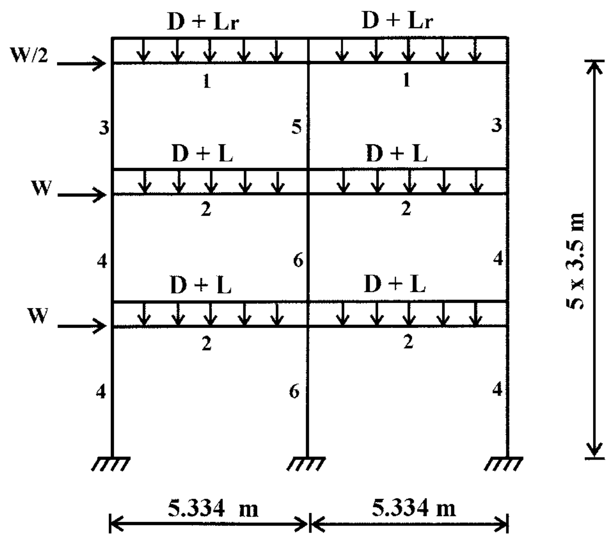

Fig. 2 3-storey, 2-bay frame

Table 2 Optimum design results of 3-storey, 2-bay frame

Group No.	Tabu Search		Simulated Annealing	
	LRFD	ASD	LRFD	ASD
1	W 18×35	W 24×55	W 16×40	W 21×50
2	W 18×35	W 21×50	W 16×40	W 18×50
3	W 8×35	W 8×31	W 14×43	W 8×40
4	W 8×35	W 8×31	W 14×43	W 14×43
5	W 10×33	W 8×40	W 12×40	W 8×40
6	W 14×48	W 10×54	W 14×43	W 14×53
Weight (kg)	3437	4202	3910	4451
Top storey drift (cm)	1.70	1.25	1.46	0.77
Max. interstorey drift (cm)	0.66	0.61	0.62	0.32
Max. interaction ratio	1.0	0.94	0.98	0.99
Computing time (min)	28.25	6.70	30.92	7.93

for ASD code. When compared the design codes, 18.2% lighter frame was obtained by LRFD than the one obtained by ASD using TS. 12.2% lighter frame was also obtained by LRFD than the one obtained by ASD while SA was considered. As regards the computing time, TS also gave lower values than SA in this example.

8.3 Design of 10-storey, single-bay frame

The last example is the 10-storey, single-bay frame which is divided into eight groups as shown in Fig. 3. The values of 6 cm and 1cm were imposed on the frame for ASD specification for the top and interstorey drift constraints. These values were calculated as 7.8 cm and 1.3 cm for LRFD specification.

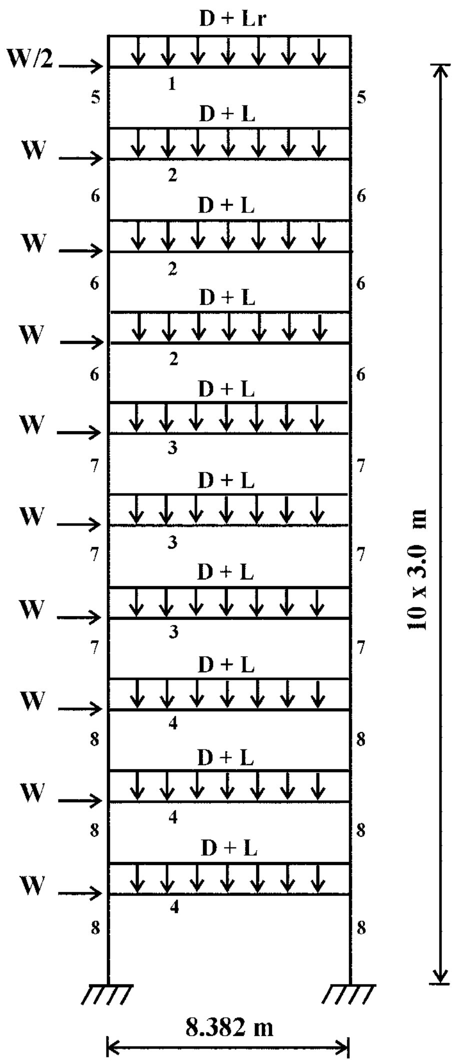

Fig. 3 10-storey, single-bay frame

For each of the four design algorithms, 10 different designs were executed again and the lightest one of those were reported in Table 3. The optimum designs were obtained after 67 and 79 cycles for LRFD and ASD respectively, when TS was considered . SA found the optimum designs at the end of 81-st and 94-th cycles for LRFD and ASD respectively.

Both displacement and stress constraints were active for LRFD code while stress constraints were only active for ASD code when TS was used. Neither stress nor displacement constraints were active for LRFD code with the use of SA. This indicates that size constraints for columns governed this design. Stress and size constraints for columns were also active for ASD code while SA was carried out.

When compared the methods, TS resulted in 1.8% and 3.9% lighter frames than the ones of the SA while LRFD and ASD codes were considered respectively.

Table 3 Optimum design results of 10-storey, single-bay frame

Group No.	Tabu search		Simulated Annealing	
	LRFD	ASD	LRFD	ASD
1	W 24×76	W 30×90	W 24×76	W 30×90
2	W 21×68	W 30×90	W 24×68	W 27×84
3	W 24×76	W 30×90	W 24×68	W 30×90
4	W 27×84	W 30×99	W 27×84	W 33×118
5	W 10×49	W 14×53	W 14×53	W 12×96
6	W 12×65	W 14×74	W 14×90	W 14×90
7	W 12×106	W 14×109	W 14×90	W 14×90
8	W 14×120	W 14×132	W 14×132	W 14×132
Weight (kg)	17750	20466	18070	21286
Top storey drift (cm)	7.80	4.55	7.61	4.01
Max. interstorey drift (cm)	1.14	0.61	1.12	0.59
Max. interaction ratio	1.0	0.97	0.89	1.0
Computing time (min)	63.25	17.21	69.80	20.18

As regards the design codes, 13.3% lighter frame was obtained by LRFD than the one obtained by ASD when TS was used. 15.1% lighter frame was also obtained by LRFD than the one obtained by ASD while SA was considered. TS spent less computing time than SA as shown in the Table 3.

9. Conclusions

The following conclusions are drawn from the design examples considered when tabu search and simulated annealing are used in the optimum design of non-linear steel frames:

1. TS obtained 1.8%-12.1% lighter frames than SA for the LRFD code. It also found 3.9%-8.2% lighter frames than SA for the ASD code. The reason for this is that TS uses more powerful techniques in searching the global optima.
2. When compared the design codes, LRFD yielded 12.1%-18.2% lighter frames than ASD for the TS method. LRFD also resulted in 12.2%-16.2% lighter frames than ASD for the SA method. The reason for this is that LRFD is based on the limit-state concept for the strength and serviceability of the structure. Load and resistance factors are determined from statistical analyses and probability theory and therefore LRFD presents more rational approach in the design of steel frames. Consequently, TS method yielded better designs together with LRFD specification.
3. As regards the computing time, TS gave lower values than SA as shown in the tables. This indicates that TS is able to find the lighter optima in lower computing time. Moreover, the computing time for LRFD is much more than the one of ASD. This is because of the multiple loadings and therefore multiple frame analyses considered in LRFD.
4. In this work, actual constraints of code specifications were used. The developed algorithms are general and can be applied to the different standards such as Eurocode or British standards.

References

Abido, M. A. (2002), "Optimal power flow using tabu search algorithm", *Electric Power Components and Systems*, **30**(5), 469-483.

Ad Hoc Committee on Serviceability Research (1986), "Structural serviceability: a critical appraisal and research needs", *J. Struct. Eng., ASCE*, **112**(12), 2646-2664.

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

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**, 1780-1795.

Barski, M. (2006), "Optimal design of shells against buckling subjected to combined loadings", *Struct. Multidiscip. Optimiz.*, **31**, 211-222.

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. (1995a), "Single and multiobjective structural optimization in discrete continuous variables using simulated annealing", *Inter. J. Numer. Methods Eng.*, **38**, 2753- 2773.

Bennage, W. A. and Dhingra, A. K. (1995b), "Optimization of truss topology using tabu search", *Inter. J. Numer. Methods Eng.*, **38**, 4035-4052.

Bland, J. A. (1995), "Discrete-variable optimal structural design using tabu search", *Struct. Optimiz.*, **10**, 87-93.

Bland, J. A. (1998a), "Structural design optimization with reliability constraints using tabu search", *Eng., Optimiz.*, **30**, 55-74.

Bland, J. A. (1998b), "A memory-based technique for optimal structural design", *Eng. Appl. Artif. Inte.*, **11**(3), 319-325.

Ceranic, B., Fryer, C. and Baines, R. W. (2001), "An application of simulated annealing to the optimum design of concrete retaining structures", *Comput. Struct.*, **79**, 1569-1581.

Chamberland, S. and Sanso, B. (2002), "Tabu search and post optimization algorithms for the update of telecommunication networks", *Canadian J. Elect. Comput. Eng.*, **27**(2), 67-74.

Chen, T. Y. and Su, J. J. (2002), "Efficiency improvement of simulated annealing in optimal structural designs", *Ad. Eng. Soft.*, **33**, 675-680.

Cunha, M. D. and Riberio, L. (2004), "Tabu search algorithms for water network optimization", *Eur. J. Operational Res.*, **157**(3), 746-758.

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

Dhingra, A. K. and Bennage, W. A. (1995b), "Topological optimization truss structures using simulated annealing", *Eng. Optimiz.*, **24**, 239-259.

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

Erdal, O. and Sonmez, F. O. (2005), "Optimum design of composite laminates for maximum buckling load capacity using simulated annealing", *Compos. Struct.*, **71**(1), 45-52.

Glover, F. (1989), "Tabu search-Part I", *ORSA Journal on Computing*, **1**(3), 190-206.

Glover, F. (1990), "Tabu search-Part II", *ORSA Journal on Computing*, **2**(1), 4-32.

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

Hasancebi, O. and Erbatur, F. (2002), "Layout optimisation of trusses using simulated annealing", *Adv. Eng. Soft.*, **33**, 681-696.

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

Jeon, Y. J. and Kim, J. C. (2004), "Application of simulated annealing and tabu search for loss minimization in distribution systems", *Inter. J. Elec. Power Energy Sys.*, **26**(1), 9-18.

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

Kumar, S., Shekbar, S. and Chakraborty, D. (2006), "Optimal pumping from skimming wells", *J. Hydrolgy Eng.*, **11**(5), 464-471.

Levy, R. and Spillers, W. R. (1994), *Analysis of Geometrically Nonlinear Structures*, Chapman and Hall, New York.

Manoharan, S. and Shanmuganathan, S. (1999), "A comparison of search mechanisms for structural optimization", *Comput. Struct.*, **73**(1-5), 363-372.

Metropolis, N., Rosenbluth, A., Teller, A. and Teller, E. (1953), "Equation of state calculations by fast computing machines", *J. Chemical Phy.*, **21**, 1087-1092.

Moita, J. M. S., Correia, V. M. F., Martins, P. G., Soares, C. M. M. and Soares, C. A. M. (2006), "Optimal design in vibration control of adaptive structures using simulated annealing algorithm", *Compo. Struct.*, **75**(1-4), 79-87.

Pai, N., Kaw, A. and Weng, M. (2003), "Optimization of laminate stacking sequence for failure load maximization using tabu search", *Composites Part B-Engineering*, **34**(4), 405-413.

Pantelidis, C. P. and Tzan, S. R. (2000), "Modified iterated annealing algorithm for structural synthesis", *Adv. Eng. Soft.*, **31**, 391-400.

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**, 1305-1316.

Rama Mohan Rao, A. and Arvind, N. (2007), "Optimal stacking sequence design of laminate composite structures using tabu embedded simulated annealing", *Struct. Eng. Mech.*, **25**(2), 239-268.

Richards, E. W. and Gunn, E. A. (2003), "Tabu search design for difficult forest management optimization problems", *Canadian J. Forest Res.*, **33**(6), 1126-1133.

Sait, S. M. and Zahra, M. M. (2002), "Tabu search based circuit optimization", *Eng. Appl. Artif. Intel.*, **15**(3-4), 357-368.

Sathiya, P., Aranvindan, S., Haq, A. N. and Panneerselvam, K. (2006), "Optimization of friction welding parameters using simulated annealing", *Indian J. Eng. Mater. Sci.*, **13**(1), 37-44.

Teegavarapu, R. S. V., Simonovic, S. P. and Slobodan, P. (2002), "Optimal operation of reservoir systems using simulated annealing", *Water Resources Management*, **16**(5), 401-428.

Zhao, F. and Zeng, X. G. (2006), "Simulated annealing-genetic algorithm for transit network optimization", *J. Comput. Civ. Eng.*, **20**(1), 57-68.

CC