

## COMPUTATIONALLY EFFICIENT OPTIMUM DESIGN OF LARGE SCALE STEEL FRAMES

S. Kazemzadeh Azad<sup>a,\*</sup>, O. Hasançebi<sup>a</sup>, S. Kazemzadeh Azad<sup>b</sup>

<sup>a</sup>*Middle East Technical University, Department of Civil Engineering, Ankara, Turkey*

<sup>b</sup>*Amirkabir University of Technology, Department of Civil and Environmental  
Engineering, Tehran, Iran*

### ABSTRACT

Computational cost of metaheuristic based optimum design algorithms grows excessively with structure size. This results in computational inefficiency of modern metaheuristic algorithms in tackling optimum design problems of large scale structural systems. This paper attempts to provide a computationally efficient optimization tool for optimum design of large scale steel frame structures to AISC-LRFD specifications. To this end an upper bound strategy (UBS), which is a recently proposed strategy for reducing the total number of structural analyses in metaheuristic optimization algorithms, is used in conjunction with an exponential variant of the well-known big bang-big crunch optimization algorithm. The performance of the UBS integrated algorithm is investigated in the optimum design of two large-scale steel frame structures with 3860 and 11540 structural members. The obtained numerical results clearly reveal the usefulness of the employed technique in practical optimum design of large-scale structural systems even using regular computers.

Received: 5 April 2014; Accepted: 26 June 2014

**KEY WORDS:** Structural optimization; metaheuristic search techniques; big bang-big crunch algorithm; upper bound strategy; large-scale steel frames; AISC-LRFD

---

\*Corresponding author: S. Kazemzadeh Azad, Department of Civil Engineering, Middle East Technical University, 06800, Ankara, Turkey

†E-mail address: saeid.azad@metu.edu.tr

## 1. INTRODUCTION

In the recent decades, development of advanced construction technologies has paved the way for extensive emergence of large scale structures. Although a general definition cannot be stated for the term large scale, however, for instance a high rise building consisting thousands of structural elements can be referred to as a large scale system. Besides the complexities involved in the construction of large structural systems, the design stage of such structures has long been recognized as one of the most challenging fields of the engineering design.

Typically, decision making about an optimum design for a large scale system requires seeking the best set of structural members which minimizes the total weight/cost of the structure while satisfying the predefined design constraints imposed by a considered design code. Generally, the main concern of structural engineers is to adopt an efficient optimization tool for optimum design of large scale systems in an acceptable computational time. In this regard, two difficulties may arise in practical applications as follows. On one hand, due to the large number of structural components, the optimum solution should be sought in a vast design space with numerous design variables. Accordingly, sometimes even finding a feasible solution satisfying all the imposed strength and serviceability constraints may become a dilemma. On the other hand, since structural analysis of a large scale system requires an excessive computational effort, an employed optimization technique may need an unacceptably extreme computational time to locate a reasonable solution [1].

In the recent years an extensive work has been conducted in developing efficient structural optimization techniques for practical applications. Generally, majority of the developed techniques are belonging to the class of stochastic search algorithms or the so called metaheuristics [2]. In spite of numerous applications of metaheuristics reported in the literature of structural optimization [3-5], due to the large number of structural analyses required by metaheuristic based techniques, the existing algorithms in their original form are not applicable to large scale systems without using expensive high performance computing techniques, such as parallel or distributed computing methods. Furthermore, since practically it is not possible to carry out numerous independent optimization runs for large scale systems, conducting modifications in the formulation of metaheuristics (e.g. parameter study) for tackling large scale problems is almost impossible in real applications. As a result, in spite of sound reputation of metaheuristic algorithms in global optimization, only a few papers have been published on optimum design of large scale steel frames using metaheuristics [1, 6].

Regarding the inherent nature of metaheuristic algorithms that need many structural analyses to find an optimum design, employing an upper bound strategy (UBS) [7] by which unnecessary structural analyses are avoided in the course of optimization can be useful. Although simple, the UBS is found to be successful in diminishing the total computational cost of a metaheuristic design optimization algorithm. Thus, investigating the performance of the UBS in optimum design of large scale structural systems can pave the way for computationally efficient optimization of such structures without employing high performance computing techniques. To this end first a robust and efficient metaheuristic technique should be adopted. Next, through integration of the UBS with the employed

optimization algorithm, its efficiency in optimum design of large scale systems can be evaluated.

In the present study as an enhanced metaheuristic optimization tool an exponential big bang-big crunch (EBB-BC) algorithm [8], which is an improved variant of the well known big bang-big crunch (BB-BC) algorithm [9], is employed in conjunction with the upper bound strategy (UBS) for tackling challenging design instances of large scale steel frames. The EBB-BC algorithm is adopted here due to its promising performance in discrete sizing optimization of steel frames under code provisions [8]. Although traditionally practical design optimization of large-scale steel frame structures is carried out typically through employing high performance computing techniques, the study attempts to facilitate such design optimizations in reasonable computational times using inexpensive regular computers. To this end, first, the UBS is integrated into the optimum design algorithm for further reduction of the number of structural analysis in the course of optimization. Next, the performance of the UBS integrated algorithm is investigated in the optimum design of two large-scale steel frame structures with 3860 and 11540 structural members according to AISC-LRFD [10] specifications; and the numerical results are discussed in detail. The remaining sections of the paper are organized as follows. The second section briefly describes the optimum design problem based on AISC-LRFD [10] specifications. In the third section the BB-BC algorithm and its exponential variant is described. The fourth section provides a detailed statement of the UBS in the design optimization process. The efficiency of the employed UBS integrated algorithm is investigated in the fifth section through large scale design examples of steel frames. A brief conclusion of the study is provided in the last section.

## 2. DISCRETE SIZING OPTIMIZATION OF STEEL FRAMES TO AISC-LRFD

This section covers the utilized design procedure based on the AISC-LRFD [10] code. In industrial applications the frame members are typically selected from a set of available steel sections which yields a discrete sizing optimization problem. For a steel frame composed of  $N_m$  members grouped into  $N_d$  design groups, the optimum design problem, based on AISC-LRFD [10] code, can be stated as follows. The objective is to find a vector of integer values  $\mathbf{I}$  (Eq. 1) representing the sequence numbers of steel sections assigned to  $N_d$  member groups

$$\mathbf{I}^T = [I_1, I_2, \dots, I_{N_d}] \quad (1)$$

to minimize the weight,  $W$ , of the structure

$$W = \sum_{i=1}^{N_d} \rho_i A_i \sum_{j=1}^{N_i} L_j \quad (2)$$

where  $A_i$  and  $\rho_i$  are the length and unit weight of the steel section selected for member

group  $i$ , respectively,  $N_i$  is the total number of members in group  $i$ , and  $L_j$  is the length of the member  $j$  which belongs to group  $i$ . Here, the objective of finding the minimum weight structure is subjected to several design constraints, including strength and serviceability requirements. According to AISC-LRFD [10] code of practice, the following relations must be satisfied for the strength requirements.

$$\left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} + \frac{8}{9} \left( \frac{M_{uxJ}}{\phi_b M_{nx}} + \frac{M_{uyJ}}{\phi_b M_{ny}} \right)_{IEL} - 1 \leq 0 \quad \text{for} \quad \left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} \geq 0.2 \quad (3)$$

$$\left[ \frac{P_{uJ}}{2\phi P_n} \right]_{IEL} + \left( \frac{M_{uxJ}}{\phi_b M_{nx}} + \frac{M_{uyJ}}{\phi_b M_{ny}} \right)_{IEL} - 1 \leq 0 \quad \text{for} \quad \left[ \frac{P_{uJ}}{\phi P_n} \right]_{IEL} < 0.2 \quad (4)$$

$$(V_{uJ})_{IEL} - (\phi_v V_n)_{IEL} \leq 0 \quad (5)$$

In Eqs. (3) to (5),  $IEL=1, 2, \dots, NEL$  is the element number,  $NEL$  is the total number of elements,  $J=1, 2, \dots, N$  is the load combination number and  $N$  is the total number of design load combinations.  $P_{uJ}$  is the required axial (tensile or compressive) strength, under  $J$ -th design load combination.  $M_{uxJ}$  and  $M_{uyJ}$  are the required flexural strengths for bending about  $x$  and  $y$ , under the  $J$ -th design load combination, respectively; where subscripts  $x$  and  $y$  are the relating symbols for strong and weak axes bending, respectively. On the other hand,  $P_n$ ,  $M_{nx}$  and  $M_{ny}$  are the nominal axial (tensile or compressive) and flexural (for bending about  $x$  and  $y$  axes) strengths of the  $IEL$ -th member under consideration.  $\phi$  is the resistance factor for axial strength, which is 0.85 for compression and 0.9 for tension (based on yielding in the gross section) and  $\phi_b$  is the resistance factor for flexure, which is equal to 0.9. Here, Eq. (5) is used for checking members' shear capacity wherein  $V_{uJ}$  is the required shear strength under  $J$ -th load combination and  $V_n$  is the nominal shear strength of the  $IEL$ -th member under consideration. In order to calculate the design shear strength the nominal shear strength is multiplied by a resistance factor  $\phi_v$  of 0.9.

In addition to the strength requirements, the serviceability criteria should be considered in the design process. The serviceability requirements considered in this research are formulated as follows:

$$\Delta_{MaxJ} - \Delta_{Max}^a \leq 0 \quad (6)$$

$$[\delta_J]_F - [\delta^a]_F \leq 0 \quad (7)$$

Eq. (6) compares the maximum lateral displacement of the structure under  $J$ -th load

combination  $\Delta_{MaxJ}$  with the maximum allowable lateral displacement  $\Delta_{Max}^a$ . Similarly, Eq. (7) checks the interstory drift of the  $F$ -th story ( $F=1, 2, \dots, NF$ ) under the  $J$ -th load combination  $[\delta_J]_S$  against the related permitted value  $[\delta^a]_S$ ; here  $NF$  is the total number of stories.

## 2.1. NOMINAL STRENGTHS

Based on AISC-LRFD [10] specification, the nominal tensile strength of a member, based on yielding in the gross section, is equal to:

$$P_n = F_y A_g \quad (8)$$

where  $F_y$  is the member's specified yield stress and  $A_g$  is the gross section of the member.

The nominal compressive strength of a member is the smallest value obtained from the limit states of flexural buckling, torsional buckling, and flexural-torsional buckling. For members with compact and/or non-compact elements, the nominal compressive strength of the member for the limit state of flexural buckling is as follows:

$$P_n = F_{cr} A_g \quad (9)$$

where  $F_{cr}$  is the critical stress based on flexural buckling of the member, calculated as:

$$\text{for } \lambda_c = \frac{Kl}{r\pi} \sqrt{\frac{F_y}{E}} \leq 1.5 \quad F_{cr} = (0.658^{\lambda_c^2}) F_y \quad (10)$$

$$\text{for } \lambda_c = \frac{Kl}{r\pi} \sqrt{\frac{F_y}{E}} > 1.5 \quad F_{cr} = \left[ \frac{0.877}{\lambda_c^2} \right] F_y \quad (11)$$

In the above equations,  $l$  is the laterally unbraced length of the member,  $K$  is the effective length factor,  $r$  is the governing radius of gyration about the axis of buckling and  $E$  is the modulus of elasticity.

The AISC-LRFD [10] code addresses the nominal compressive strength based on the limit state of torsional and flexural-torsional buckling, for doubly symmetric members with compact and/or non-compact elements. For this limit state, Eq. (9) is still applicable with the following modifications:

$$\text{for } \lambda_e \leq 1.5 \quad F_{cr} = (0.658^{\lambda_e^2}) F_y \quad (12)$$

$$\text{for } \lambda_e > 1.5 \quad F_{cr} = \left[ \frac{0.877}{\lambda_e^2} \right] F_y \quad (13)$$

where

$$\lambda_e = \sqrt{F_y / F_e} \quad (14)$$

$$F_e = \left[ \frac{\pi^2 EC_w}{(K_z l_z)^2} + GJ \right] \frac{1}{I_x + I_y} \quad (15)$$

In Eq. (15),  $C_w$  is the warping constant,  $G$  is the shear modulus,  $J$  is the torsional constant,  $I_x$  and  $I_y$  are moments of inertia about principal axes,  $l_z$  is the unbraced length for torsional buckling, and  $K_z$  is the effective length factor for torsional buckling. In this study  $K_z$  is conservatively taken as unity.

The nominal flexural strength of a member is the minimum value obtained according to the limit states of yielding, lateral-torsional buckling, flange local buckling, and web local buckling. The flexural capacity based on the limit state of yielding is as follows:

$$M_n = M_p = ZF_y \leq 1.5SF_y \quad (16)$$

where  $Z$  is the plastic modulus and  $S$  is the section modulus of the member for the axis of bending. For doubly symmetric sections, the flexural capacity considering the limit state of lateral-torsional buckling is as follows:

$$M_n = \begin{cases} M_p & \text{if } L_b \leq L_p \\ C_b \left[ M_p - (M_p - M_r) \left( \frac{L_b - L_p}{L_r - L_p} \right) \right] \leq M_p & \text{if } L_p < L_b \leq L_r \\ M_{cr} \leq M_p & \text{if } L_b > L_r \end{cases} \quad (17)$$

where  $L_b$  is the laterally unbraced length of the member,  $L_p$  is the limiting laterally unbraced length for full plastic bending capacity,  $L_r$  is the limiting laterally unbraced length for inelastic lateral-torsional buckling,  $M_r$  is the limiting buckling moment, and  $M_{cr}$  is the critical elastic moment for the lateral-torsional buckling. The modification factor for non-uniform moment diagram,  $C_b$ , is defined by Eq. (18),

$$C_b = \frac{12.5 M_{\max}}{2.5 M_{\max} + 3 M_A + 4 M_B + 3 M_C} \quad (18)$$

where  $M_{\max}$ ,  $M_A$ ,  $M_B$ , and  $M_C$  are absolute values of maximum moment, moment at quarter point, centerline, and three-quarter point of the unbraced segment, respectively.

The nominal flexural strength of members with doubly symmetric sections and non-compact flanges, considering the limit state of flange local buckling, is given below:

$$M_n = \begin{cases} M_p & \text{if } \lambda_f \leq \lambda_p^f \\ M_p - (M_p - M_r^f) \left( \frac{\lambda_f - \lambda_p^f}{\lambda_r^f - \lambda_p^f} \right) & \text{if } \lambda_p^f < \lambda_f \leq \lambda_r^f \\ M_{cr}^f \leq M_p & \text{if } \lambda_f > \lambda_r^f \end{cases} \quad (19)$$

where  $\lambda_f$  is the flange slenderness parameter,  $\lambda_p^f$  is the limiting value of  $\lambda_f$  for full plastic bending capacity,  $\lambda_r^f$  is the limiting value of  $\lambda_f$  for inelastic flange local buckling,  $M_r^f$  is the limiting moment for flange buckling, and  $M_{cr}^f$  is the critical elastic moment for flange local buckling.

The nominal flexural strength of members with doubly symmetric sections and non-compact webs, considering the limit state of flange web buckling, is given below:

$$M_n = \begin{cases} M_p & \text{if } \lambda_w \leq \lambda_p^w \\ M_p - (M_p - M_r^w) \left( \frac{\lambda_w - \lambda_p^w}{\lambda_r^w - \lambda_p^w} \right) & \text{if } \lambda_p^w < \lambda_w \leq \lambda_r^w \end{cases} \quad (20)$$

where  $\lambda_w$  is the web slenderness parameter,  $\lambda_p^w$  is the limiting value of  $\lambda_w$  for full plastic bending capacity,  $\lambda_r^w$  is the limiting value of  $\lambda_f$  for inelastic web local buckling,  $M_r^w$  is the limiting moment for web buckling, and  $M_{cr}^w$  is the critical elastic moment for web local buckling.

The nominal shear strength of unstiffened webs of doubly symmetric members, subjected to shear in the plane of the web, is as follows:

$$\text{for } h/t_w \leq 418/\sqrt{F_{yw}} \quad V_n = 0.6F_{yw} A_w \quad (21)$$

$$\text{for } 418/\sqrt{F_{yw}} < h/t_w \leq 523/\sqrt{F_{yw}} \quad V_n = 0.6F_{yw} A_w (418/\sqrt{F_{yw}}) / (h/t_w) \quad (22)$$

$$\text{for } 523/\sqrt{F_{yw}} < h/t_w \leq 260 \quad V_n = 132000 A_w / (h/t_w)^2 \quad (23)$$

where  $h$  is the clear distance between flanges less the fillet or corner radius for rolled shapes,  $t_w$  is the web thickness,  $A_w$  is the shear area, and  $F_{yw}$  is the yield stress of the web in ksi; also  $V_n$  in Eq. (23) is in ksi. Here, to keep the original formulation of the code, Eqs.

(21) to (23) are presented in British units. For members subjected to shear perpendicular to the plane of the web, the nominal shear strength is calculated through Eq. (21) as well.

## 2.2. EFFECTIVE LENGTH FACTOR

In order to calculate the nominal compressive strength, the effective length factor,  $K$ , should be determined for each member. This factor can be computed using the frame buckling monograph developed by Jackson and Moreland [11]. For sway frames, the effective length factor for columns is computed as follows:

$$\frac{\alpha^2 G_i G_j - 36}{6(G_i + G_j)} = \frac{\alpha}{\tan \alpha} \quad (24)$$

$$G_i = \frac{\sum I_{ci}/l_{ci}}{\sum I_{bi}/l_{bi}}, \quad G_j = \frac{\sum I_{cj}/l_{cj}}{\sum I_{bj}/l_{bj}} \quad (25)$$

where  $\alpha = \pi/K$ ,  $i$  and  $j$  subscripts correspond to end- $i$  and end- $j$  of the compression member, and subscripts  $c$  and  $b$ , in building structures, refer to columns and beams connecting to the joint under consideration, respectively. Parameters  $I$  and  $l$  in the above equations, represent the moment of inertia and unbraced length of the member, respectively. Here,  $K$  factor for beam, bracings and non-sway column elements is taken as 1.

## 3. OPTIMIZATION ALGORITHM

The big bang-big crunch (BB-BC) optimization algorithm is a novel metaheuristic technique based on the BB-BC theory of the universe evolution [9]. Numerous engineering optimization applications of the algorithm have been reported in the literature so far [12-21]. This section outlines the main steps in the implementation of a standard BB-BC algorithm as follows.

**Step1. Initial population:** Form an initial population through randomly spreading individuals (candidate solutions) over all the search space (first big bang) in a uniform manner. This step is applied once.

**Step 2. Evaluation:** The initial population is evaluated, where structural analyses of all the individuals are carried out with the set of steel sections adopted for design variables, and force and deformation responses are obtained under the loads. The objective function values of the feasible individuals that satisfy all the problem constraints are directly computed from Eq. (2). However, infeasible individuals that violate some of the problem constraints are penalized using an external penalty function approach, and their objective function values are computed according to Eq. (26) [22].

$$f = W \left[ 1 + p \left( \sum_i c_i \right) \right] \quad (26)$$



In Eq. (26),  $f$  is the constrained objective function value,  $c_i$  is  $i$ -th problem constraint violation and  $p$  is the penalty coefficient used to tune the intensity of penalization as a whole. This parameter is generally set to an appropriate static value of  $p = 1$ . The fitness scores of the individuals are then calculated by taking the inverse of their objective function values (i.e. fitness =  $1/W$  or  $1/f$  for feasible and infeasible solutions, respectively). The fitness scores are assigned as the mass values for the individuals.

**Step 3. Big crunch phase:** Calculate the center of mass by taking the weighted average using the coordinates (design variables) and the mass values of every single individual or choose the fittest individual amongst all as their center of mass (the latter approach is used in the present study).

**Step 4. Big bang phase:** Generate new individuals by using normal distribution (big bang phase). For a continuous variable optimization problem, Eq. (27) is used at each iteration to generate new solutions around the center of mass.

$$x_i^{new} = x_i^c + \alpha \cdot N(0,1)_i \frac{(x_i^{\max} - x_i^{\min})}{k} \quad (27)$$

where  $x_i^c$  is the value of  $i$ -th continuous design variable in the fittest individual,  $x_i^{\min}$  and  $x_i^{\max}$  are the lower and upper bounds on the value of  $i$ -th design variable, respectively,  $N(0,1)_i$  is a random number generated according to a standard normal distribution with mean ( $\mu$ ) zero and standard deviation ( $\sigma$ ) equal to one,  $k$  is the iteration number, and  $\alpha$  is a constant.

However, when a discrete set of available sections is used for sizing the frame members, Eq. (28) is employed to round off the real values to the nearest integers representing the sequence number of available sections in a given section list.

$$I_i^{new} = I_i^c + \text{round} \left[ \alpha \cdot N(0,1)_i \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (28)$$

where  $I_i^c$  is the value of  $i$ -th discrete design variable in the fittest individual, and  $I_i^{\min}$  and  $I_i^{\max}$  are its lower and upper bounds, respectively.

**Step 5. Elitism:** Keep the fittest individual found so far in a separate place or as a member of the population.

**Step 6. Termination:** Go to Step 2 until a stopping criterion is satisfied, which can be imposed as a maximum number of iterations or no improvement of the best design over a certain number of iterations.

Recently, in order to enhance the performance of the BB-BC algorithm in discrete design optimization, Eq. (29) is proposed in Ref. [8] as a new formulation in lieu of Eq. (28). In the new formulation the use of  $n$ -th power ( $n \geq 2$ ) of a random number  $r_i$  is motivated using any appropriate statistical distribution, which may not be necessarily a normal distribution.

$$I_i^{new} = I_i^c + \text{round} \left[ \alpha \cdot r_i^n \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (29)$$

The rationale behind Eq. (29) is to achieve a satisfactory trade-off between the exploration and exploitation characteristics of the BB-BC algorithm. Accordingly, Eq. (30) provided in Ref. [8] is an instance of Eq. (29), referred to as the exponential BB-BC (EBB-BC) algorithm, where the use of an exponential distribution in conjunction with the third power of random number is preferred.

$$I_i^{new} = I_i^c \pm \text{round} \left[ \alpha \cdot \text{Exp}(\lambda = 1)_i^3 \frac{(I_i^{\max} - I_i^{\min})}{k} \right] \quad (30)$$

While using the EBB-BC algorithm for discrete optimization, it is noted that sometimes after a certain number of iterations, no new solutions are generated; i.e. the subsequent individuals simply replicate the former ones. As a remedy to this situation, the following routine proposed in Ref. [8] is integrated into the algorithm to make sure that a new solution will differ from the former one at least by one variable.

```

Set  $\sigma := 1.0$ ;
Quitloop:= False;
Repeat
  Generate  $\mathbf{I}^{new}$  from  $\mathbf{I}^c$  using Eq. (30)
  If  $\mathbf{I}^{new} = \mathbf{I}^c$  then Quitloop := true
  else  $\sigma := \sigma + 1.0$ 
Until Quitloop;

```

Accordingly, if all the design variables in a newly generated solution remain unchanged after applying Eq. (30), the generation process is iterated in the same way by increasing the standard deviation of normal distribution  $\sigma$  by one every time till a different design is produced. Apparently, the increased standard deviation facilitates occurrence of larger changes in the generated individuals.

#### 4. UBS IN METAHEURISTIC BASED DESIGN OPTIMIZATION

The UBS [7] is a simple yet efficient technique which can be utilized in conjunction with all metaheuristic algorithms that employ a  $\mu + \lambda$  selection scheme in their algorithmic models. This selection scheme is first characterized by the well-known variant of evolution strategies (ES) technique referred to as  $(\mu + \lambda)$ -ES in the literature [23]. Typically, at each generation of the  $(\mu + \lambda)$ -ES,  $\mu$  parents generate  $\lambda$  offspring; and then a deterministic selection is performed by selecting the  $\mu$  best individuals out of  $\mu$  parents and  $\lambda$  offspring in reference to the individuals' fitness scores [24]. This way, the number of individuals to produce the next generation is reduced back to  $\mu$  every time. It should be noted that the evolutionary scheme employed in the EBB-BC algorithm works on the basis of the same

principle. At each iteration of the EBB-BC algorithm  $\mu = 1$  parent generates  $\lambda$  candidate solutions ( $\lambda=50$  in this study), and only one individual survives out of  $\lambda + 1$  solutions after implementing the selection and elitism schemes.

In the EBB-BC algorithm discussed in the previous section, the current best design found during the optimization process is used to generate new candidate solutions for the next iteration. Then, every candidate solution generated is subjected to evaluation such that a conventional structural analysis is first carried out per design and then its penalized weight (which is the base of comparison between the solutions) is computed through the application of Eq. (26). The idea behind the UBS is to impose the current best design as the upper bound for the forthcoming candidates to eliminate unnecessary structural analysis and associated fitness computations for those candidates that have no chance of surpassing the best solution. Basically, the key factor in this approach is to define the penalized weight of the current best solution found during the previous iterations as an upper bound for the net weight of the newly generated candidate solutions. Thus, any new candidate solution with a net weight greater than this upper bound will not be analyzed and this will lessen the computational burden of the optimization algorithm.

The pseudo-code for the utilized UBS integrated optimization algorithm is outlined below, where *ite\_cnt* and *ite\_max* stand for the current and maximum iteration numbers, respectively.

```

Repeat
  Generate  $\mathbf{I}_i$  ( $i:=1,..,\lambda$ ) from  $\mathbf{I}_{best}$  using Eq. (30)
  For  $i:=1$  to  $\lambda$  do
    begin
      Calculate  $W(\mathbf{I}_i)$ 
      If  $W(\mathbf{I}_i) \leq f(\mathbf{I}_{best})$ 
        then
          Perform structural analysis of  $\mathbf{I}_i$ 
          Compute  $P(\mathbf{I}_i)$  and  $f(\mathbf{I}_i)$ 
          If  $f(\mathbf{I}_i) \leq f(\mathbf{I}_{best})$ 
            then
              Update the upper bound  $f(\mathbf{I}_{best}) = f(\mathbf{I}_i)$ 
            end
          else
            Activate UBS
            Eliminate  $\mathbf{I}_i$ 
          end
        end
      Set new  $\mathbf{I}_{best}$ 
       $ite\_cnt := ite\_cnt + 1;$ 
    until  $ite\_cnt > ite\_max$ 

```

In the aforementioned strategy,  $\lambda$  number of candidate solutions  $\mathbf{I}_i$  are first generated from the current best design in a usual manner, i.e. through application of Eq. (30). Then, in the first step the net weight  $W(\mathbf{I}_i)$  of each candidate solution is calculated only; not the penalized weight. This computation is straightforward and can be done with a trivial computational effort. If a candidate solution has a net weight  $W(\mathbf{I}_i)$  smaller than or equal to the penalized weight of the current best design  $f(\mathbf{I}_{best})$ , the structural analysis of the candidate solution is processed and its penalized weight is computed. In the opposite case, i.e.  $W(\mathbf{I}_i) > f(\mathbf{I}_{best})$ , however, the UBS is activated and the candidate solution is automatically removed from the population without undergoing structural analysis phase for response computations, since such a candidate is unlikely to improve the current best design  $\mathbf{I}_{best}$ .

As reflected in the pseudo-code, it is noticed that the upper bound value can be dynamically modified during the analysis of the individuals of a population as well. In this study the employed UBS updates the upper bound value dynamically with respect to the penalized weight of each individual immediately after it is analyzed. In this strategy if the penalized weight of the individual is less than the current upper bound value, then the upper bound value is updated to the lower value i.e. the penalized weight of the individual.

It is apparent that, at each iteration of the UBS integrated optimization algorithm the number of analyzed individuals is not necessarily equal to the population size. It is worth mentioning that, for further increasing the efficiency of the UBS during the design optimization process, the structural analysis of the candidate designs is accomplished in an order based on their net weights. In other words, for a population of candidate designs the structural analyses of candidate designs with smaller net weights are performed prior to those have larger net weights. This can further increase the probability of excluding the remaining individuals from the structural analysis stage without changing the algorithmic structure of the utilized strategy. The main concern here is the amount of reduction in computational effort using the abovementioned approach. This is investigated in the next section through design examples of large scale steel frames.

## 5. NUMERICAL EXAMPLES

In this section the efficiency of the UBS in reducing the number of structural analyses is investigated through two design optimization examples of large scale frames. To this end, the optimization algorithm is coded in MATLAB [25] and employed in conjunction with SAP2000 v14.1 [26] structural analysis package using application programming interface (API) for analysis and design of structural systems sampled during the course of optimization process.

The population size of the algorithm is set to 50 and the value of parameter  $\alpha$  in Eq. (30) is selected as 0.25. The value of penalty constant  $p$  is taken as 1. Further, the maximum number of iterations (ite\_max) is considered as the termination criterion. This is set to 500 iterations for both the examples. The material properties of steel are taken as follows:

modulus of elasticity ( $E$ ) = 200 GPa, yield stress ( $F_y$ ) = 248.2 Mpa, and unit weight of the steel ( $\rho$ ) = 7.85 ton/m<sup>3</sup>.

In the following examples the UEBB-BC notation is used to refer to the UBS integrated optimization algorithm. It should be underlined that since the efficiency and robustness of the EBB-BC algorithm is recently demonstrated by the authors in Ref. [8], an improvement of the algorithm in terms of quality of the optimum solution located is not intended in the present study. Instead, the aim of the study is to accelerate computational efficiency of the sizing optimization by reducing the computing cost through a smaller number of structural analyses.

### 5.1. EXAMPLE 1: 3860-MEMBER STEEL FRAME

The 20-story steel frame shown in Figure 1 is adopted as the first design example. The frame is composed of 3860 structural members, including 1836 beam, 1064 column and 960 bracing elements. The stability of structure is provided through moment resisting connections as well as X-type bracing systems along the  $x$  and  $y$  directions. Considering practical fabrication requirements, the 3860 members of the frame are collected under 73 member groups. The member grouping is performed in both plan and elevation levels. In elevation level the structural members are grouped in every two stories. In plan level, columns are considered in 5 different column groups ( $CG_1$  through  $CG_5$ ) as depicted in Figure 2; beams are divided into two groups as outer and inner beams; and bracings are assumed to be in one group. Therefore, based on both elevation and plan level groupings, there are totally 43 column groups, 20 beam groups, and 10 bracing groups considered as 73 sizing design variables in this example. It is worth mentioning that floor slabs shown in Figure 1(a) are just for better illustration of the structure; and are not modeled in the analysis stage. For design purpose, the frame is subjected to the following 10 load combinations according to ASCE 7-98 [27]:

- (1)  $1.4D$
- (2)  $1.2D + 1.6L$
- (3)  $1.2D + 1.0E_x + 0.5L$
- (4)  $1.2D + 1.0E_{ex} + 0.5L$
- (5)  $1.2D + 1.0E_y + 0.5L$
- (6)  $1.2D + 1.0E_{ey} + 0.5L$
- (7)  $0.9D + 1.0E_x$
- (8)  $0.9D + 1.0E_{ex}$
- (9)  $0.9D + 1.0E_y$
- (10)  $0.9D + 1.0E_{ey}$

where  $D$  and  $L$  denote the dead and live loads, respectively;  $E_x$  and  $E_y$  are the earthquake loads applied to the center of mass in  $x$  and  $y$  directions, respectively;  $E_{ex}$  and  $E_{ey}$  are the earthquake loads applied considering the effect of accidental eccentricity of the center of mass in  $x$  and  $y$  directions, respectively. Based on ASCE 7-98 [27] the amount of eccentricity is set to 5% of the dimension of the structure perpendicular to the direction of the applied earthquake load.

The live loads acting on the floor and roof beams are 10 and 7 kN/m, respectively. In the case of dead loads, besides the uniformly distributed loads of 14 and 12 kN/m applied on floor and roof beams, respectively, the self-weight of the structure is also considered.

The earthquake loads, are calculated based on the equivalent lateral force procedure outlined in ASCE 7-98 [27]. Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.1W_s$  where  $W_s$  is the total dead load of the building. The computed base shear is distributed to each floor based on the following equation:

$$F_x = \frac{w_x h_x^k V}{\sum_{i=1}^n w_i h_i^k} \quad (31)$$

where  $F_x$  is the induced lateral seismic force at level  $x$ ;  $w$  is portion of the total gravity load assigned to the related level (i.e. level  $i$  or  $x$ ); and  $h$  is the height from base to the related level. Here,  $k$  is determined based on the structure period. It is equal to 1 for structures with a period of 0.5 sec or less; and 2 for structures with a period of 2.5 sec or more. For structures with a period in range of 0.5 to 2.5 sec,  $k$  is calculated through linear interpolation [27]. It is worth mentioning that the period of the structure is calculated using the following equation given in ASCE 7-98 [27].

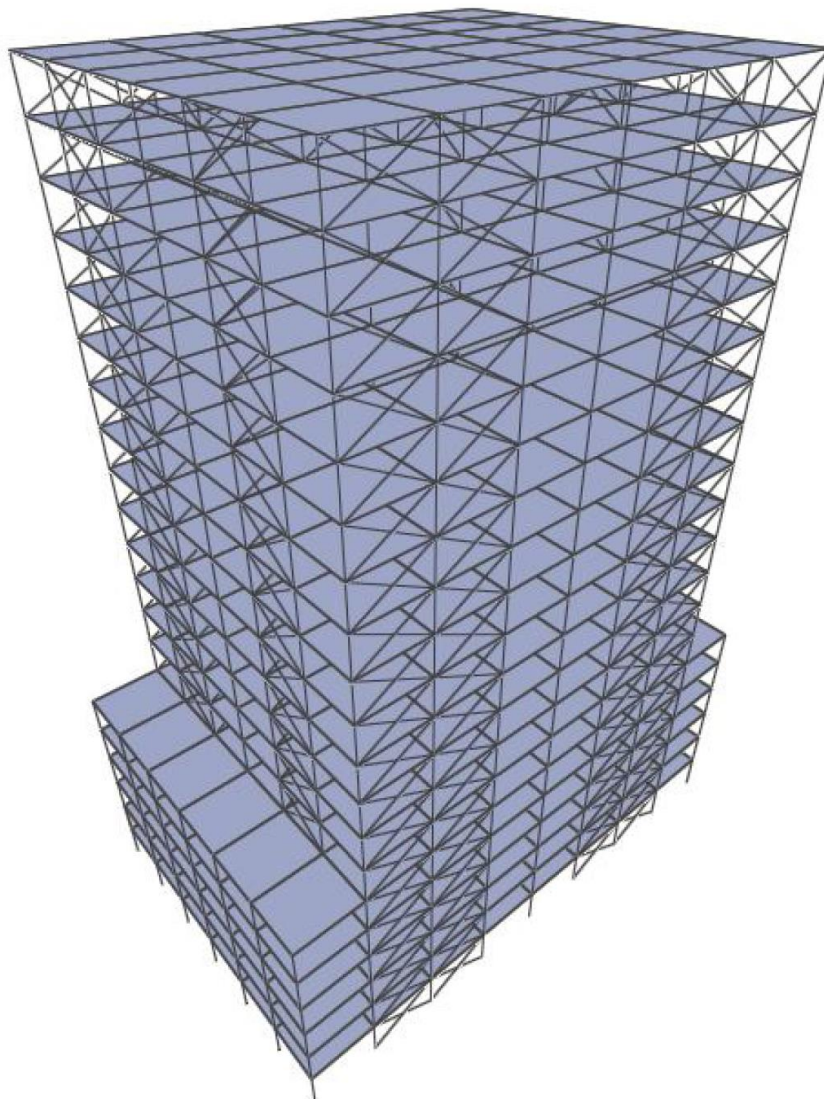
$$T = C_T h_n^{3/4} \quad (32)$$

where  $C_T$  is taken as 0.0488 and  $h_n$  is the height of the building; namely 70 m for this example. Hence, the period of the structure,  $T$ , is 1.181 sec. Based on the obtained period the value of parameter  $k$  in Eq. (31) is taken as 1.341 for this example. It should be noticed that since the self-weight of the structure changes during the course of optimization, apparently, the values of dead and earthquake loads change accordingly.

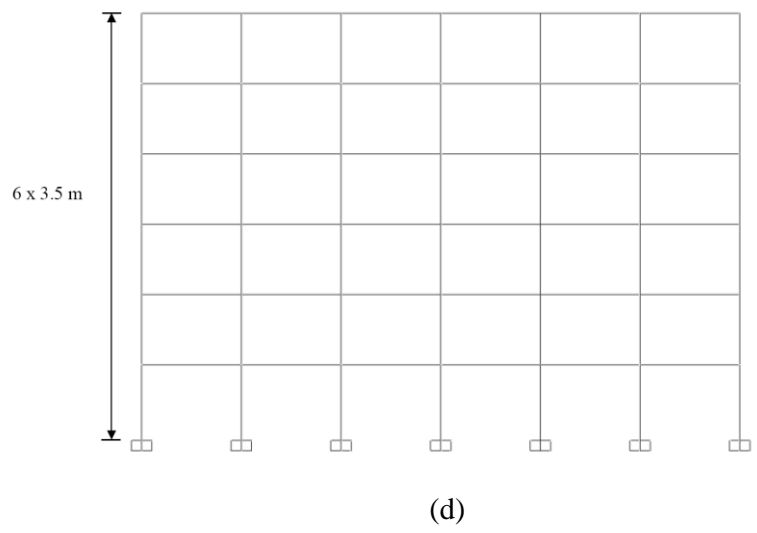
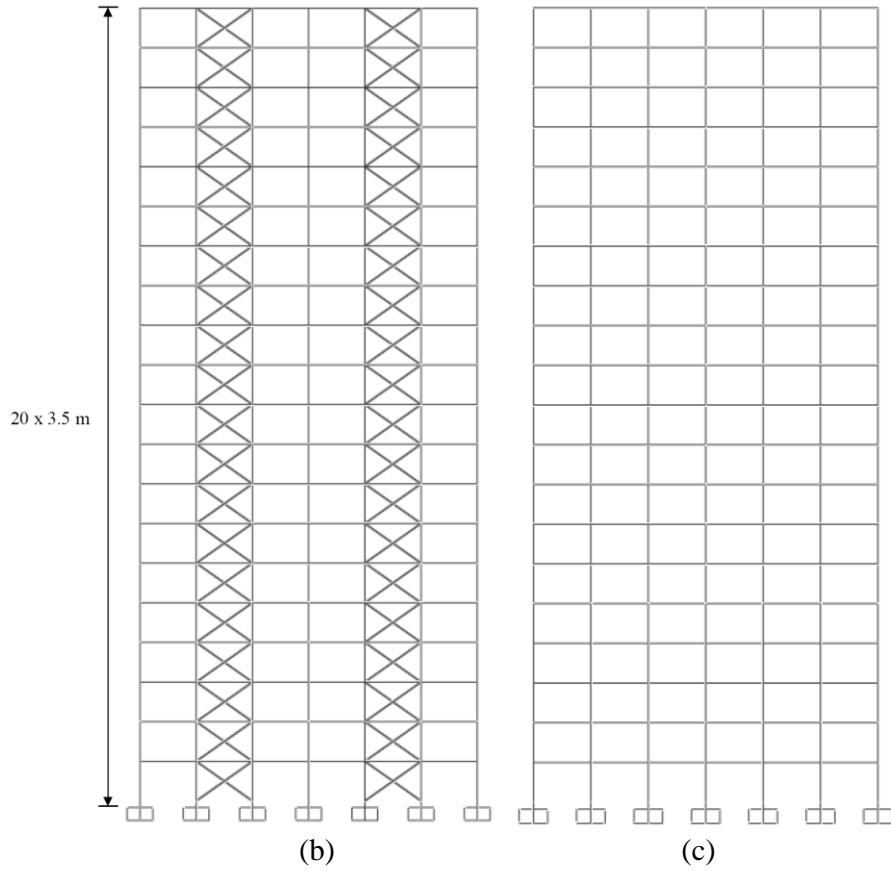
The beam elements are continuously braced along their lengths by the floor system; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , for buckling of columns as well as beams and bracings is taken as 1. The maximum lateral displacement of the top story is limited to 0.18 m and the upper limit of interstory drift is taken as  $h/400$ , where  $h$  is the story height. The interstory drifts are calculated based on the displacement of center of mass of each story. The maximum lateral displacement of the top story is calculated with respect to the maximum displacements of the ends of the structure. Here, horizontal displacements of all joints of each story are constrained to each other based on a rigid diaphragm assumption. For this example, the wide-flange (W) profile list composed of 268 ready sections is used to size the structural members.

Optimum desing of the frame is carried out using the UEbb-BC algorithm and the results are tabulated in Table 1. The algorithm is executed until the termination condition, which is the maximum number of iterations, i.e. 500, is met. The optimization history of the frame is presented in Figure 3, which shows the variation of the penalized weight of the

current best design obtained so far in the search process. Here, the number of structural analyses performed in the UBS integrated algorithm is calculated by counting candidate designs that undergo structural analysis. For the original algorithm, i.e. EBB-BC, this can be simply obtained through multiplying the total number of iterations ( $ite\_cnt$ ) by the population size  $\lambda$ . Bearing in mind that a population size of  $\lambda = 50$  is employed over a maximum number of 500 iterations ( $ite\_max = 500$ ), the number of structural analyses performed by the EBB-BC algorithm (without UBS) is equal to 25000. However, as tabulated in Table 1, when UEBB-BC algorithm is employed, it is found that only 9979 structural analyses are required for sizing the frame. In this example, 15021 unnecessary analyses are avoided as a result of employing the UEBB-BC algorithm.



(a)





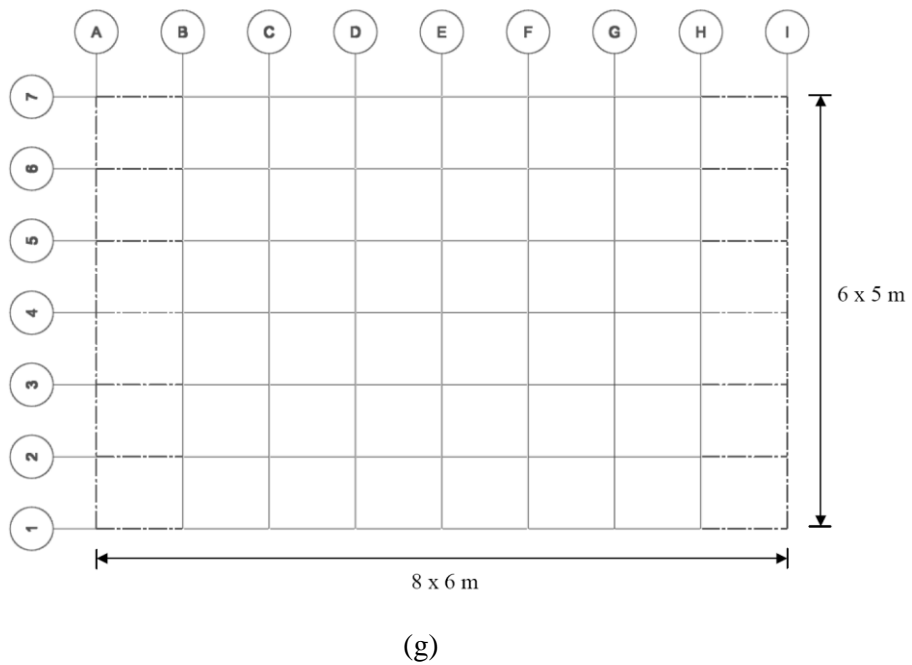
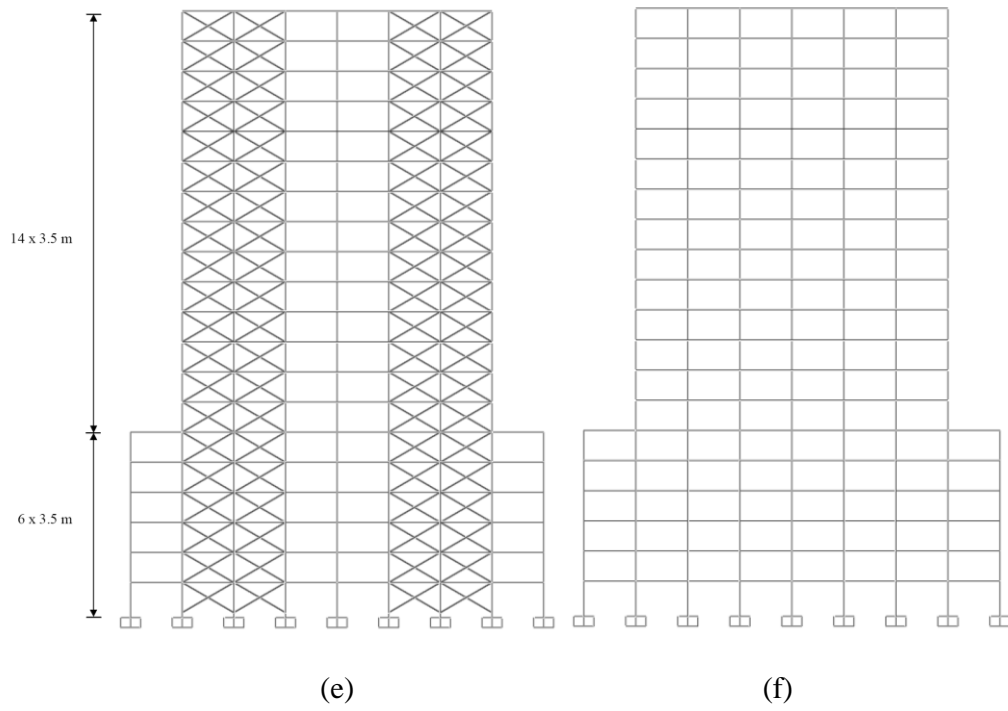


Figure 1: 3860-member steel frame, (a) 3-D view (b) side view of frames B, D, F, and H (c) side view of frames C, E, and G (d) side view of frames A, and I (e) side view of frames 1, 3, 5, and 7 (f) side view of frames 2, 4, and 6 (g) plan view.

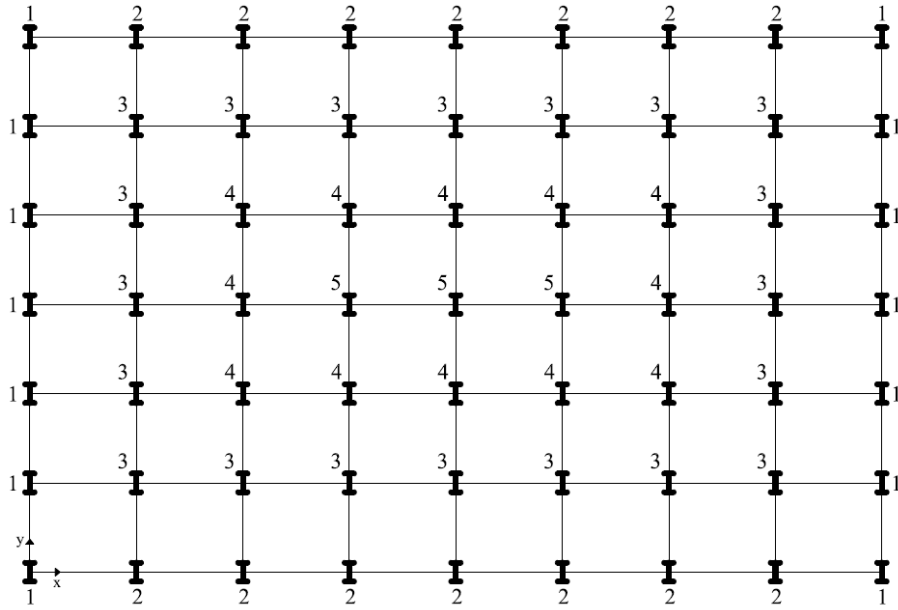


Figure 2: Column grouping in plan level for 3860-member steel frame

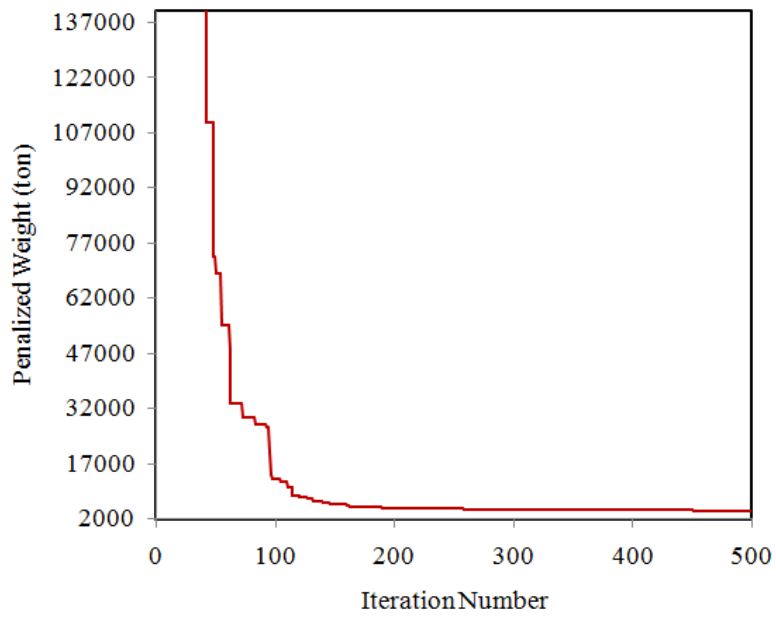


Figure 3: Optimization history of 3860-member steel frame example

Table 1: Optimum design obtained for 3860-member steel frame

Stories	Groups	Ready Section	Stories	Groups	Ready Section
1-2	CG <sub>1</sub> *	W27X146	11-12	CG <sub>1</sub>	N.A.
	CG <sub>2</sub>	W12X210		CG <sub>2</sub>	W30X99
	CG <sub>3</sub>	W36X359		CG <sub>3</sub>	W12X305
	CG <sub>4</sub>	W40X593		CG <sub>4</sub>	W30X261
	CG <sub>5</sub>	W8X67		CG <sub>5</sub>	W44X230
	IB*	W40X183		IB	W44X230
	OB*	W14X22		OB	W40X149
	BR*	W40X167		BR	W16X77
3-4	CG <sub>1</sub>	W40X167	13-14	CG <sub>1</sub>	N.A.
	CG <sub>2</sub>	W12X230		CG <sub>2</sub>	W16X89
	CG <sub>3</sub>	W36X650		CG <sub>3</sub>	W40X174
	CG <sub>4</sub>	W24X335		CG <sub>4</sub>	W16X100
	CG <sub>5</sub>	W14X68		CG <sub>5</sub>	W10X100
	IB	W16X26		IB	W14X43
	OB	W40X235		OB	W36X135
	BR	W12X53		BR	W33X152
5-6	CG <sub>1</sub>	W24X335	15-16	CG <sub>1</sub>	N.A.
	CG <sub>2</sub>	W27X178		CG <sub>2</sub>	W12X79
	CG <sub>3</sub>	W27X539		CG <sub>3</sub>	W44X262
	CG <sub>4</sub>	W36X439		CG <sub>4</sub>	W24X250
	CG <sub>5</sub>	W30X99		CG <sub>5</sub>	W33X263
	IB	W44X230		IB	W30X132
	OB	W14X26		OB	W14X68
	BR	W12X72		BR	W21X62
7-8	CG <sub>1</sub>	N.A.	17-18	CG <sub>1</sub>	N.A.
	CG <sub>2</sub>	W27X368		CG <sub>2</sub>	W24X117
	CG <sub>3</sub>	W18X234		CG <sub>3</sub>	W40X167
	CG <sub>4</sub>	W33X221		CG <sub>4</sub>	W36X245
	CG <sub>5</sub>	W40X321		CG <sub>5</sub>	W30X292
	IB	W12X72		IB	W40X149
	OB	W33X130		OB	W33X141
	BR	W12X72		BR	W14X48
9-10	CG <sub>1</sub>	N.A.	19-20	CG <sub>1</sub>	N.A.
	CG <sub>2</sub>	W30X326		CG <sub>2</sub>	W36X300
	CG <sub>3</sub>	W14X455		CG <sub>3</sub>	W18X60
	CG <sub>4</sub>	W12X120		CG <sub>4</sub>	W44X230
	CG <sub>5</sub>	W18X86		CG <sub>5</sub>	W10X54
	IB	W27X94		IB	W21X50
	OB	W44X230		OB	W30X116
	BR	W24X84		BR	W24X62
Weight (ton)					4117.43
No. analyses required without UBS					25000
No. analyses performed					9979
No. saved analyses					15021

\*CG denotes column group with respect to Figure 2, IB: inner beams, OB: outer beams, BR: bracings

## 5.2. EXAMPLE 2: 11540-MEMBER STEEL FRAME

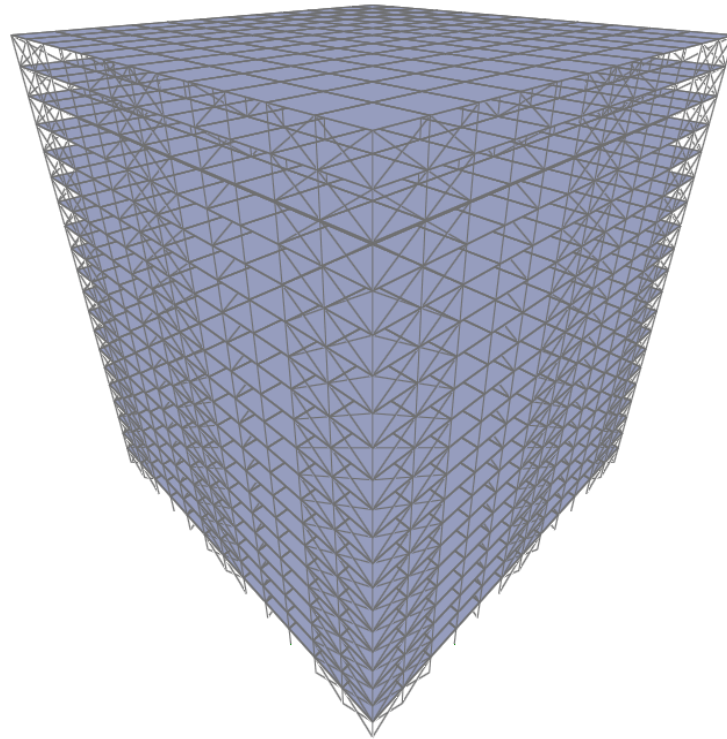
The 20-story steel frame depicted in Figure 4 is selected as the second example. The frame, which is one of the largest steel frame instances investigated so far, is composed of 11540 structural members, including 6240 beam, 3380 column and 1920 bracing elements. The stability of structure is provided through moment resisting connections as well as X-type bracing systems along the  $x$  and  $y$  directions. Considering practical fabrication requirements, the 11540 members of the frame are collected under 100 member groups. The member grouping is performed in both plan and elevation levels. In elevation level the structural members are grouped in every two stories. In plan level, columns are considered in 7 different column groups (CG<sub>1</sub> through CG<sub>7</sub>) as depicted in Figure 6 (where all columns located on each square are treated as one column group); beams are divided into two groups as outer and inner beams; and bracings are assumed to be in one group. Therefore, based on both elevation and plan level groupings, there are totally 70 column groups, 20 beam groups, and 10 bracing groups considered as 100 sizing design variables in this example. For the sake of clarity, columns' orientations are shown in Figure 5. It is worth mentioning that floor slabs shown in Figure 4(a) are just for better illustration of the structure; and are not modeled in the analysis stage.

For design purpose, the frame is subjected to the same 10 load combinations described in the first example. The live loads acting on the floor and roof beams are 12 and 7 kN/m, respectively. In the case of dead loads, besides the uniformly distributed loads of 15 and 12 kN/m applied on floor and roof beams, respectively, the self-weight of the structure is also considered. The earthquake loads, are calculated based on the same procedure described in the first example. Here, the resulting seismic base shear ( $V$ ) is taken as  $V = 0.1W_s$  where  $W_s$  is the total dead load of the building. Further, in Eq. (32),  $C_T$  is taken as 0.0488 and  $h_n$  is 70 m. Hence, the period of the structure,  $T$ , is computed as 1.181 sec. Based on the obtained period the value of parameter  $k$  in Eq. (31) is taken as 1.341 for this example.

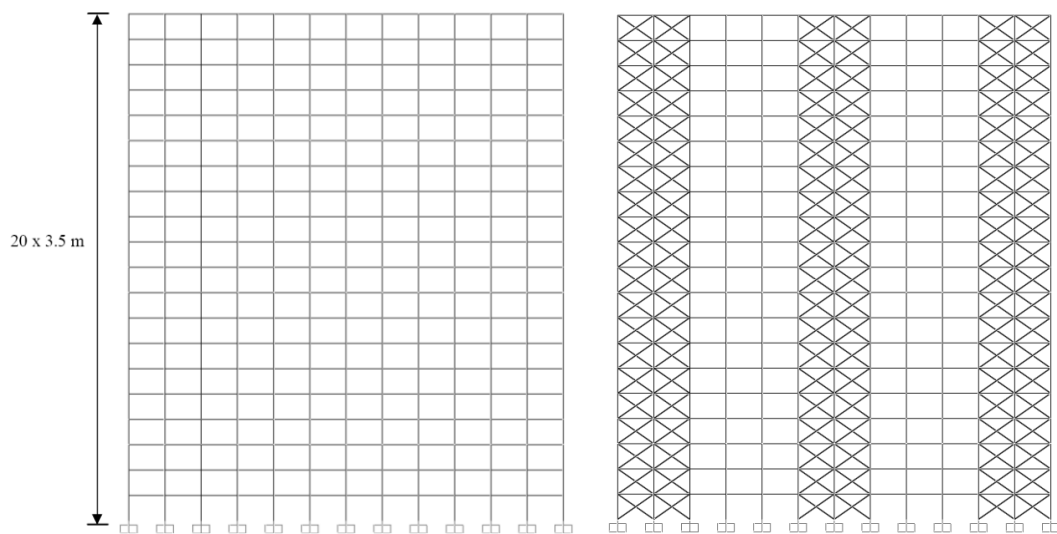
The beam elements are continuously braced along their lengths by the floor system; and columns and bracings are assumed to be unbraced along their lengths. The effective length factor,  $K$ , for buckling of columns as well as beams and bracings is taken as 1. The maximum lateral displacement of the top story is limited to 0.18 m and the upper limit of interstory drift is taken as  $h/400$ , where  $h$  is the story height. Here, the wide-flange (W) profile list composed of 162 ready sections (W16 through W44) is used to size the structural members.

Optimum design of the 11540-member frame is performed using the UEBB-BC algorithm and the results are given in Table 2. The algorithm is executed until the termination condition, which is the maximum number of iterations, i.e. 500, is met. The optimization history of the frame is presented in Figure 7. Bearing in mind that a population size of  $\lambda = 50$  is employed over a maximum number of 500 iterations ( $ite\_max = 500$ ), the number of structural analyses performed by the EBB-BC algorithm (without UBS) is equal to 25000. However, as presented in Table 2, when UEBB-BC algorithm is employed, it is found that only 7616 structural analyses are needed. Here, 17384 unnecessary analyses are avoided as a result of employing the UEBB-BC algorithm. The numerical results attained in the investigated examples clearly indicate the fruitfulness of integrating the UBS with

modern optimization techniques (e.g. EBB-BC) for computationally efficient optimum design of large scale steel frames.

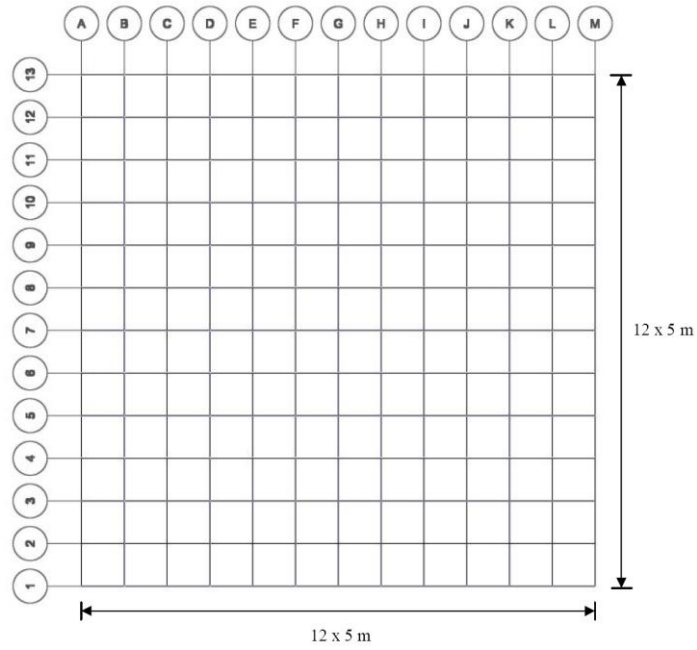


(a)



(b)

(c)



(d)

Figure 4: 11540-member steel frame, (a) 3-D view (b) side view of frames 2, 3, 4, 6, 7, 8, 10, 11, 12, B, C, D, F, G, H, J, K and L (c) side view of frames 1, 5, 9, 13, A, E, I, and M (d) plan view

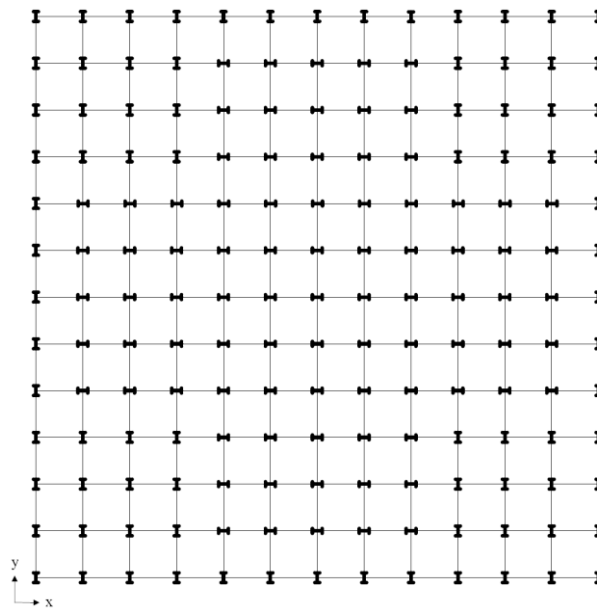


Figure 5: Columns' orientations of 11540-member steel frame

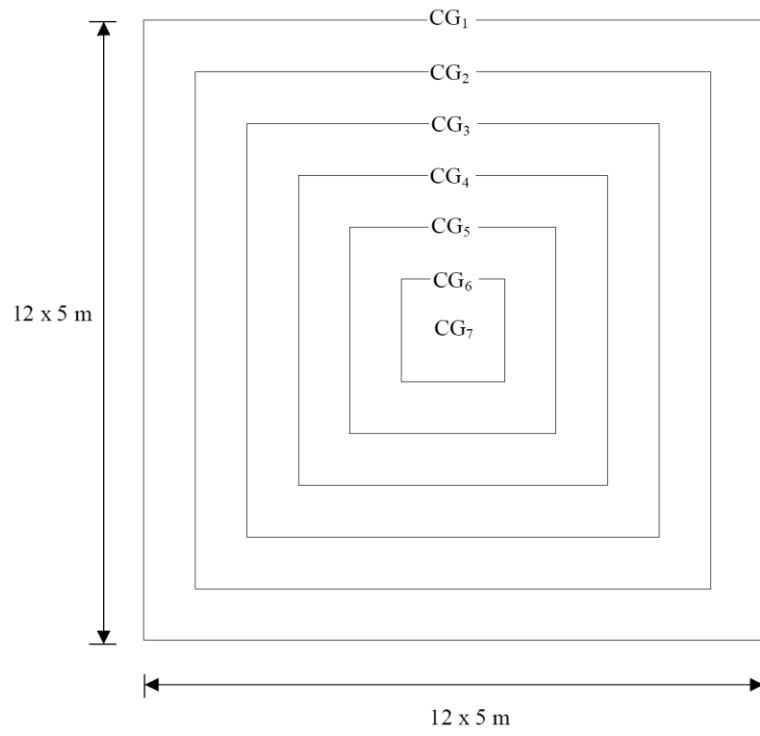


Figure 6: Outline of column grouping in plan level for 11540-member steel frame

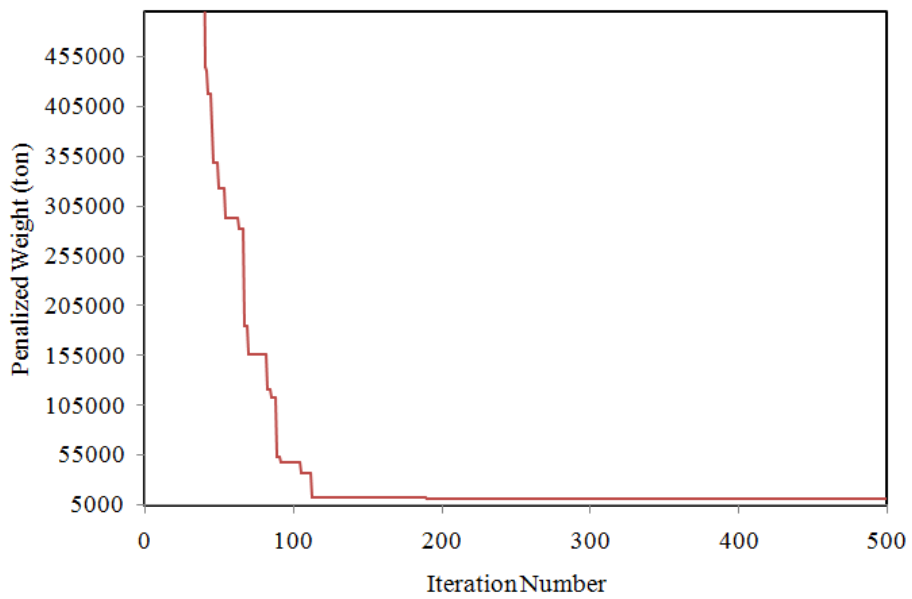


Figure 7: Optimization history of 11540-member steel frame example

Table 2: Optimum design obtained for 11540-member steel frame

Stories	Groups	Ready Section	Stories	Groups	Ready Section
1-2	CG <sub>1</sub> *	W36X393	11-12	CG <sub>1</sub>	W24X408
	CG <sub>2</sub>	W40X249		CG <sub>2</sub>	W33X118
	CG <sub>3</sub>	W36X393		CG <sub>3</sub>	W18X234
	CG <sub>4</sub>	W40X297		CG <sub>4</sub>	W36X328
	CG <sub>5</sub>	W44X335		CG <sub>5</sub>	W30X235
	CG <sub>6</sub>	W18X258		CG <sub>6</sub>	W36X170
	CG <sub>7</sub>	W40X431		CG <sub>7</sub>	W27X146
	IB*	W18X40		IB	W40X199
	OB*	W21X57		OB	W33X291
	BR*	W18X211		BR	W21X62
3-4	CG <sub>1</sub>	W24X250	13-14	CG <sub>1</sub>	W24X250
	CG <sub>2</sub>	W40X372		CG <sub>2</sub>	W33X118
	CG <sub>3</sub>	W44X262		CG <sub>3</sub>	W30X191
	CG <sub>4</sub>	W36X256		CG <sub>4</sub>	W40X264
	CG <sub>5</sub>	W40X277		CG <sub>5</sub>	W21X93
	CG <sub>6</sub>	W27X258		CG <sub>6</sub>	W33X241
	CG <sub>7</sub>	W33X291		CG <sub>7</sub>	W16X89
	IB	W16X26		IB	W27X84
	OB	W18X40		OB	W33X291
	BR	W33X130		BR	W27X129
5-6	CG <sub>1</sub>	W44X290	15-16	CG <sub>1</sub>	W24X117
	CG <sub>2</sub>	W18X234		CG <sub>2</sub>	W24X84
	CG <sub>3</sub>	W44X335		CG <sub>3</sub>	W36X256
	CG <sub>4</sub>	W36X245		CG <sub>4</sub>	W36X232
	CG <sub>5</sub>	W33X241		CG <sub>5</sub>	W16X89
	CG <sub>6</sub>	W18X258		CG <sub>6</sub>	W33X130
	CG <sub>7</sub>	W44X290		CG <sub>7</sub>	W40X167
	IB	W40X174		IB	W24X62
	OB	W40X199		OB	W18X60
	BR	W27X94		BR	W21X68
7-8	CG <sub>1</sub>	W21X182	17-18	CG <sub>1</sub>	W18X192
	CG <sub>2</sub>	W33X152		CG <sub>2</sub>	W24X117
	CG <sub>3</sub>	W18X175		CG <sub>3</sub>	W33X118
	CG <sub>4</sub>	W33X291		CG <sub>4</sub>	W44X262
	CG <sub>5</sub>	W27X178		CG <sub>5</sub>	W44X335
	CG <sub>6</sub>	W40X199		CG <sub>6</sub>	W40X297
	CG <sub>7</sub>	W24X250		CG <sub>7</sub>	W36X160
	IB	W36X135		IB	W36X150
	OB	W24X68		OB	W24X62
	BR	W40X174		BR	W21X62
9-10	CG <sub>1</sub>	W40X372	19-20	CG <sub>1</sub>	W18X119
	CG <sub>2</sub>	W33X169		CG <sub>2</sub>	W33X152
	CG <sub>3</sub>	W24X250		CG <sub>3</sub>	W24X76
	CG <sub>4</sub>	W30X261		CG <sub>4</sub>	W16X36
	CG <sub>5</sub>	W24X335		CG <sub>5</sub>	W16X36



Stories	Groups	Ready Section	Stories	Groups	Ready Section
	CG <sub>6</sub>	W27X178		CG <sub>6</sub>	W18X143
	CG <sub>7</sub>	W27X448		CG <sub>7</sub>	W30X108
	IB	W24X55		IB	W24X55
	OB	W40X277		OB	W40X167
	BR	W27X94		BR	W18X40
Weight (ton)					10756.63
No. analyses required without UBS					25000
No. analyses performed					7616
No. saved analyses					17384

\*CG denotes column group with respect to Figure 6, IB: inner beams, OB: outer beams, BR: bracings

## 6. CONCLUDING REMARKS

An upper bound strategy (UBS) is utilized for computationally efficient optimum design of large scale steel frame structures. In order to investigate the efficiency of the strategy, the EBB-BC optimization algorithm is integrated with the UBS for optimum design of large scale steel frames to AISC-LRFD [10] specifications. Based on the employed UBS, the upper bound limit for the net weights of the newly generated candidate solutions is dynamically updated, resulting in a considerable reduction in computational cost of the optimization process. The numerical results obtained through discrete sizing optimization of two large scale steel frames with 3860 and 11540 structural members clearly reveal that the UBS is capable of reducing the computational effort required to approach a reasonable design in large scale optimization applications. It can be deduced that the UBS integrated optimization techniques can be efficiently utilized as alternative methods to the expensive high-performance computing techniques; and can pave the way for practical optimum design of large scale structures using inexpensive computers.

## REFERENCES

1. Hasançebi O., Bahçecioğlu T., Kurç Ö., Saka, MP. Optimum design of high-rise steel buildings using an evolution strategy integrated parallel algorithm. *Comput Struct*, 2011; **89**: 2037–2051.
2. Yang X-S. Nature-inspired metaheuristic algorithms, Luniver Press, 2008.
3. Lamberti L, Pappalettere C. Metaheuristic design optimization of skeletal structures: a review. *Computational Technology Reviews*, 2011; 1–32.
4. Saka MP. Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review, in Topping BHV, (Editor), *Civil Engineering Computations: Tools and Techniques*, Saxe-Coburg Publications, Stirlingshire, UK, 2007; 105–147.
5. Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. *Adv Eng Software*, 2013; **59**: 19–28.
6. Hasançebi O. Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. *Comput Struct*, 2008; **86**: 119–132.

7. Kazemzadeh Azad S, Hasaebi O, Kazemzadeh Azad S. Upper Bound Strategy for Metaheuristic Based Design Optimization of Steel Frames. *Adv Eng Software*, 2013; **57**: 19–32.
8. Hasaebi O, Kazemzadeh Azad S. An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. *Comput Struct*, 2012; **110–111**: 167–179.
9. Erol OK, Eksin, I. A new optimization method: Big Bang-Big Crunch, *Adv. Eng. Software*, 2006; **37**(2): 106–111.
10. American Institute of Steel Construction (AISC). Manual of Steel Construction, Load & Resistance Factor Design. 2nd ed. Chicago, 1994.
11. McGuire W. Steel Structures. Prentice-Hall, 1968.
12. Afshar MH, Motaei I. Constrained Big Bang-Big Crunch algorithm for optimal solution of large scale reservoir operation problem. *Int. J. Optim. Civil Eng.*, 2011; **1**(2): 357–375.
13. Tang H, Zhou J, Xue S, Xie L. Big Bang-Big Crunch optimization for parameter estimation in structural systems, *Mech. Syst. Signal Proc.*, 2010; **24**(8): 2888–2897.
14. Camp CV. Design of space trusses using big bang-big crunch optimization, *J. Struct. Eng.*, ASCE, 2007; **133**: 999–1008.
15. Kaveh A., Abbasgholiha H. Optimum design of steel sway frames using big bang–big crunch algorithm, *Asian J. Civ. Eng.*, 2011; **12**: 293–317.
16. Lamberti L, Pappalettere C. A fast big bang-big crunch optimization algorithm for weight minimization of truss structures, In: Tsompanakis Y, Topping BHV, editors. Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Stirlingshire, UK, Civil-Comp Press; 2011.
17. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang-Big Crunch algorithm. *Comput. Struct.*, 2009; **87**(17-18): 1129–1140.
18. Kaveh A, Talatahari S. Optimal design of Schwedler and ribbed domes via hybrid Big Bang-Big Crunch algorithm. *J. Construct. Steel Res.*, 2009; **66** (3): 412–419.
19. Kaveh A, Talatahari S. A discrete big bang-big crunch algorithm for optimal design of skeletal structures, *Asian J. Civ. Eng.*, 2010; **11**: 103–123.
20. Kaveh A, Talatahari S, Sheikholeslami R. Optimum seismic design of gravity retaining walls using the heuristic big bang-big crunch algorithm, In: Tsompanakis Y, Topping BHV, editors. Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Civil-Comp Press, Stirlingshire, UK, Civil-Comp Press; 2011.
21. S. Kazemzadeh Azad, O. Hasaebi, O.K. Erol, Evaluating efficiency of big bang-big crunch algorithm in benchmark engineering optimization problems, *Int. J. Optim. Civ. Eng.*, 2011; **1**: 495–505.
22. Hasaebi O, arbas S, Dogan E, Erdal F, Saka MP. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Comput Struct*, 2009; **87**: 284–302.
23. Schwefel H-P. Numerical optimization of computer models. Chichester, UK, John Wiley & Sons, 1981.
24. Michalewicz Z. Genetic algorithms + data structures = evolution programs, 3rd ed.,

Springer-Verlag, 1996.

25. MATLAB. Version 7.8, The MathWorks Inc., 3 Apple Hill Drive, Natick.
26. SAP2000. Version 14.1, Computers and Structures Inc., Berkeley, CA.
27. ASCE 7-98, Minimum design loads for buildings and other structures: Revision of ANSI/ASCE 7-95, American Society of Civil Engineers, 2000.