Chapter 6
Harmony Search Algorithms in Structural Engineering

M.P. Saka, I. Aydogdu, O. Hasancebi, and Z.W. Geem

Abstract. Harmony search method is widely applied in structural design optimization since its emergence. These applications have shown that harmony search algorithm is robust, effective and reliable optimization method. Within recent years several enhancements are suggested to improve the performance of the algorithm. Among these Mahdavi has presented two versions of harmony search methods. He named these as improved harmony search method and global best harmony search method. Saka and Hasancebi (2009) have suggested adaptive harmony search where the harmony search parameters are adjusted automatically during design iterations. Coelho has proposed improved harmony search method. He suggested an expression for one of the parameters of standard harmony search method. In this chapter, the optimum design problem of steel space frames is formulated according to the provisions of LRFD-AISC (Load and Resistance Factor Design-American Institute of Steel Corporation). The weight of the steel frame is taken as the objective function to be minimized. Seven different structural optimization algorithms are developed each of which are based on one of the above mentioned versions of harmony search method. Three real size steel frames are

M.P. Saka · I. Aydogdu
Department of Engineering Sciences,
Middle East Technical University, Ankara, Turkey
e-mail: mpsaka@metu.edu.tr, aydogdu@metu.edu.tr

O. Hasancebi
Department of Civil Engineering,
Middle East Technical University, Ankara, Turkey
e-mail: oguzhan@metu.edu.tr

Z.W. Geem
Information Technology Program,
iGlobal University, Annandale, Virginia, USA
e-mail: zwgeem@gmail.com

X.-S. Yang & S. Koziel (Eds.): Computational Optimization & Applications, SCI 359, pp. 145–182.
springerlink.com

designed using each of these algorithms. The optimum designs obtained by these techniques are compared and performance of each version is evaluated.

Keywords: Structural Optimization, Metaheuristic Techniques, Harmony Search Algorithm.

6.1 Introduction

Building construction causes use of large amount of land, consumption of energy and water. Furthermore, production of the materials used in the building construction add large amount of pollution into the atmosphere. It is important to reduce the amount of natural resources utilized in building construction if we would like to reduce the environmental impact and have a sustainable development. Green construction is becoming a common practice all over the world which intends to construct buildings that are environmentally friendly and resource-efficient throughout its life cycle. This practice covers all the stages from setting to design, construction and operation. Overdesigns and use of excessive materials are not desired because they consume more of natural sources and add more pollution to the atmosphere. Hence it is clear that in order to have a sustainable development structures are required to be designed and built using sufficient amount of material but not more. Structural design optimization tools exactly try to achieve this goal. They aim to design the steel structures such that the steel frame has the minimum weight and in the mean time the response of the frame under the external loads that the frame may be subjected to during its life time is within the design code limitations. Design of steel structures has its own features and not similar to the design of other structures. Designer cannot use any section she/he may desire but to select among the set of steel profiles available in practice for beams and columns of the frame under consideration. This selection is required to be carried out such that the frame with the selected steel profiles should have the displacements less than those prescribed in the design code and its members have sufficient strength to satisfy the strength limitations under the external loads. In the mean time its cost is the minimum.

In this chapter firstly the design optimization problem of steel space frames according to the provisions of LRFD-AISC (Load and Resistance factor Design-American Institution of Steel Corporation) [1] is presented. The weight of the steel frame is taken as the objective function to be minimized. Such formulation of the optimum design problem yields a discrete programming problem. The solution of this programming problem is obtained by harmony search algorithm [2-6]. This method is one of the recent combinatorial optimization techniques that belong to general class of what is called metaheuristic algorithms. Metaheuristic algorithms [7-11] finds the solution of optimization problems by utilizing certain tactics that are generally inspired from the nature, though not limited to, instead of classical procedures that move along the descending direction of gradient of objective function. Harmony search method mimics music improvisation. Harmony search method is widely applied in structural design optimization since its emergence [12-19]. These applications have shown that harmony search algorithm is robust, effective and reliable optimization method. Within recent years several enhancements are suggested to improve the performance of the harmony search method. Among these Mahdavi [20, 21] has presented two versions of harmony search methods. He named them as improved harmony search method and global best harmony search method. Hasancebi et. al. (2010) [22, 23] has suggested adaptive harmony search where the harmony search parameters are adjusted automatically during design iterations. Coelho [24] has proposed improved harmony search method. He suggested an expression for one the parameters of standard harmony search method. In this chapter seven different structural optimization algorithms are developed each of which is based on one of the above mentioned versions of harmony search method. Three steel space frames are designed using each of these algorithms. The optimum designs obtained by each of these techniques are compared and performance of each version is evaluated.

6.2 Discrete Optimum Design of Steel Space Frames to LRFD-AISC

The design of steel space frames necessitates the selection of steel sections for its columns and beams from a standard steel section tables such that the frame satisfies the serviceability and strength requirements specified by the code of practice while the economy is observed in the overall or material cost of the frame. When the design constraints are implemented from LRFD-AISC the following discrete programming problem is obtained.

6.2.1 The Objective Function

The objective function is taken as the minimum weight of the frame which is expressed as in the following.

$$\textit{Minimize} \quad W = \sum_{r=1}^{ng} m_r \sum_{s=1}^{t_r} \ell_s \tag{6.1}$$

where; W defines the weight of the frame, m_r is the unit weight of the steel section selected from the standard steel sections table that is to be adopted for group r. t_r is the total number of members in group r and ng is the total number of groups in the frame. l_s is the length of member s which belongs to group r.

6.2.2 Strength Constraints

For the case where the effect of warping is not included in the computation of the strength capacity of W-sections that are selected for beam-column members of the frame the following inequalities given in Chapter H of LRFD-AISC are required to be satisfied.

$$\text{for}\frac{P_u}{\phi P_n} \geq 0.2\,;\, g_{s,i} = \frac{P_u}{\phi P_n} + \frac{8}{9}\left(\frac{M_u}{\phi_b M_{nx}} + \frac{M_u}{\phi_b M_{ny}}\right) \leq 1{,}0 \tag{6.2}$$

$$\text{for}\frac{P_u}{\phi P_n} \geq 0.2\,;\, g_{s,i} = \frac{P_u}{2\phi\, P_n} + \left(\frac{M_u}{\phi_b M_{nx}} + \frac{M_u}{\phi_b M_{ny}}\right) \leq 1{,}0 \tag{6.3}$$

where, M_{nx} is the nominal flexural strength at strong axis (x axis), M_{ny} is the nominal flexural strength at weak axis (y axis), M_{ux} is the required flexural strength at strong axis (x axis), M_{uy} is the required flexural strength at weak axis (y axis), P_n is the nominal axial strength (tension or compression) and P_u is the required axial strength (tension or compression) for member i. l represents the loading case. The values of M_{ux} and M_{uy} are required to be obtained by carrying out $P - \Delta$ analysis of the steel frame. This is an iterative process which quite time consuming. In Chapter C of LRFD-AISC an alternative procedure is suggested for the computations of M_{ux} and M_{uy} values. In this procedure, two first order elastic analyses are carried out. In the first, frame is analyzed under the gravity loads only where the sway of the frame is prevented to obtain M_{nt} values. In the second, the frame is analyzed only under the lateral loads to find M_{lt} values. These moment values are then combined using the following equation as given in the design code.

$$M_u = B_1 M_{nt} + B_2 M_{lt} \tag{6.4}$$

where B_1 is the moment magnifier coefficient and B_2 is the sway moment magnifier coefficient. The details of how these coefficients are calculated are given in Chapter C of LRFD-AISC [1].

Eqns. (6.2) and (6.3) represents strength constraints for doubly and singly symmetric steel members subjected to axial force and bending. If the axial force in member k is tensile force the terms in these equations are given as: P_{uk} is the required axial tensile strength, P_{nk} is the nominal tensile strength, ϕ becomes ϕ_t in the case of tension and called strength reduction factor which is given as 0.90 for yielding in the gross section and 0.75 for fracture in the net section, ϕ_b is the strength reduction factor for flexure given as 0.90, M_{uxk} and M_{uyk} are the required flexural strength M_{nxk} and M_{uyk} are the nominal flexural strength about major and minor axis of member k respectively. It should be pointed out that required flexural bending moment should include second-order effects. LRFD suggests an approximate procedure for computation of such effects which is explained in C1 of LRFD. In the case the axial force in member k is compressive force the terms in Eqns. (6.2) and (6.3) are defined as: P_{uk} is the required compressive strength, P_{nk} is the nominal compressive strength, and ϕ becomes ϕ_c which is the resistance factor for compression given as 0.85. The remaining notations in Eqns. (6.16) and (6.17) are the same as the definition given above.

The nominal tensile strength of member k for yielding in the gross section is computed as $P_{nk} = F_y A_{gk}$ where F_y is the specified yield stress and A_{gk} is the gross area of member k. The nominal compressive strength of member k is computed as

$P_{nk} = A_{gk}F_{cr}$ where $F_{cr} = \left(0.658^{\lambda_c^2}\right)F_y$ for $\lambda_c \le 1.5$ and $F_{cr} = \left(0.877/\lambda_c^2\right)F_y$ for $\lambda_c > 1.5$ and $\lambda_c = \frac{Kl}{r\pi}\sqrt{F_y/E}$. In these expressions E is the modulus of elasticity, K and l are the effective length factor and the laterally unbraced length of member k respectively.

6.2.3 Displacement Constraints

The lateral displacements and deflection of beams in steel frames are limited by the steel design codes due to serviceability requirements. According to the ASCE Ad Hoc Committee report [25], the accepted range of drift limits in the first-order analysis is 1/750 to 1/250 times the building height H with a recommended value of H/400. The typical limits on the inter-story drift are 1/500 to 1/200 times the story height. Based on this report the deflection limits recommended are proposed in [26, 27, 28] for general use which is repeated in Table 6.1.

Table 6.1 Displacement limitations for steel frames

	Item	Deflection Limit
1	Floor girder deflection for service live load	L/360
2	Roof girder deflection	L/240
3	Lateral drift for service wind load	H/400
4	Inter-story drift for service wind load	H/300

6.2.3.1 Deflection Constraints

It is necessary to limit the mid-span deflections of beams in a steel space frame not to cause cracks in brittle finishes that they may support due to excessive displacements. Deflection constraints can be expressed as an inequality limitation as shown in the following.

$$g_{dj} = \frac{\delta_{jl}}{\delta_j^u} - 1 \le 0 \quad j = 1,\ldots,n_{sm}, \quad l = 1,\ldots,n_{lc} \tag{6.5}$$

where, δ_{jl} is the maximum deflection of j^{th} member under the l^{th} load case, δ_j^u is the upper bound on this deflection which is defined in the code as span/360 for beams carrying brittle finishers, n_{sm} is the total number of members where deflections limitations are to be imposed and n_{lc} is the number of load cases.

6.2.3.2 Drift Constraints

These constraints are of two types. One is the restriction applied to the top story sway and the other is the limitation applied on the inter-story drift.

Top Story Drift Constraint

Top story drift limitation can be expressed as an inequality constraint as shown in the following.

$$g_{td\,j} = \frac{\left(\Delta_{top}\right)_{jl}}{H\,/\,Ratio} - 1 \le 0 \qquad j = 1,\ldots,n_{jtop}\,, \quad l = 1,\ldots,n_{lc} \tag{6.6}$$

where H is the height of the frame, n_{jtop} is the number of joints on the top story, n_{lc} is the number of load cases, $(\Delta_{top})_{jl}$ is the top story drift of the j^{th} joint under l^{th} load case. Ratio is a constant value given in ASCE Ad Hoc Committee report [25].

Inter-story Drift

In multi-story steel frames the relative lateral displacements of each floor is required to be limited. This limit is defined as the maximum inter-story drift which is specified as h_{sx}/Ratio where h_{sx} is the story height and Ratio is a constant value given in ASCE Ad Hoc Committee report [25].

$$g_{id\,j} = \frac{\left(\Delta_{oh}\right)_{jl}}{h_{sx}\,/\,Ratio} - 1 \le 0 \qquad j = 1,\ldots.,n_{st}\,, \quad l = 1,\ldots\ldots.n_{lc} \tag{6.7}$$

where n_{st} is the number of story, n_{lc} is the number of load cases and $(\Delta_{oh})_{jl}$ is the story drift of the j^{th} story under l^{th} load case.

6.2.4 Geometric Constraints

In steel frames it is not desired that column section for upper floor should not have a larger section than the lower story column for practical reasons. Because having a larger section for upper floor requires a special joint arrangement which is neither preferred nor economical. The same applies to the beam-to-column connections. The W-section selected for any beam should have a flange width smaller than or equal to the flange width of the W-section selected for the column to which the beam is to be connected. These are shown in Fig. 6.1 and named as geometric constraints. These limitations are included in the design optimization model to satisfy practical requirements. Two types of geometric constraints are considered in the mathematical model. These are column-to-column geometric constraints and beam-to-column geometric limitations.

6.2.4.1 Column-to-Column Geometric Constraints

The depth and the unit weight of W sections selected for the columns of two consecutive stores should be either equal to each other or the one in the upper story should be smaller than the one in the lower story. These limitations are included in the design problem as inequality constraints as shown in the following.

$$g_{cdi} = \frac{D_i}{D_{i-1}} - 1 \leq 0 \quad i = 2......n_j \tag{6.8}$$

$$g_{cmi} = \frac{m_i}{m_{i-1}} - 1 \leq 0 \quad i = 2......n_j \tag{6.9}$$

where n_j is the number of stories, m_i is the unit weight of W section selected for column story i, m_{i-1} is the unit weight of W section selected for of column story (i–1), D_i is the depth of W section selected for of column story i and D_{i-1} is the depth of W section selected for of column story (i-1).

6.2.4.2 Beam-to-Column Geometric Constraints

When a beam is connected to a flange of a column, the flange width of the beam should be less than or equal to the flange width of the column so that the connection can be made without difficulty. In order to achieve this, the flange width of the beam should be less than or equal to $(D - 2t_b)$ of the column web dimensions in the connection where D and t_b are the depth and the flange thickness of W section respectively as shown in Fig. 6.1.

$$g_{bci} = \frac{(B_f)_{bi}}{D_{ci} - 2(t_{bc})_i} - 1 \leq 0 \qquad i = 1......n_{j1} \tag{6.10}$$

or

$$g_{bbi} = \frac{(B_f)_{bi}}{(B_f)_{ci}} - 1 \leq 0 \quad i = 1......n_{j2} \tag{6.11}$$

where n_{j1} is the number of joints where beams are connected to the web of a column, n_{j2} is the number of joints where beams connected to the flange of a column, D_{ci} is the depth of W section selected for the column at joint i, $(t_{bc})_i$ is the flange thickness of W section selected for the column at joint i, $(B_f)_{ci}$ is the flange width of W section selected for the column at joint i and $(B_f)_{bi}$ is the flange width of W section selected for the beam at joint i.

The above optimum design of steel space frames problem where the objective function is given in Eqn. (6.1) and the constraints are described from Eqn. (6.2) to Eqn. (6.11) is a combinatorial optimization problem of discrete optimization. This is because the solution of the problem necessitates the selection of appropriate steel sections for the beams and columns of the frame from W-sections list such that the objective function described in Eqn. (6.1) has the minimum value while the design constraints given in inequalities from Eqn. (6.2) to Eqn. (6.11) are satisfied. Designer has to find out the suitable combination of W-sections that makes the frame weight minimum in the same time the design code provisions are all satisfied. Here the selection of a W-section from an available steel profile list is carried out by choosing an integer number from a set which consist of integer numbers starting 1 to the total number of sections in the list. This integer number is the sequence number of that particular W-section. Hence the design solution is a set of integer numbers each of which represents the sequence number of W-section in the design pool. This is a combinatorial optimization problem [7,8].

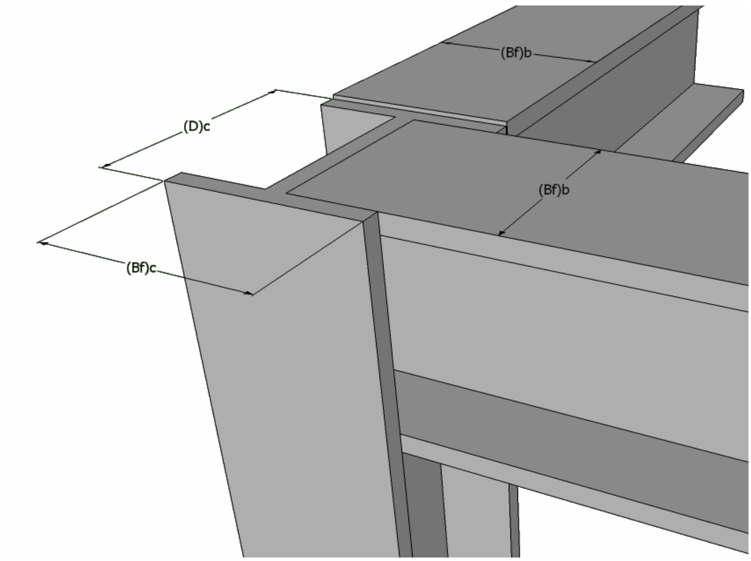


Fig. 6.1 Beam column geometric constraints

In last two decades a new kind of algorithms have emerged which make use of certain heuristics in order to explore a search space efficiently. These methods are called metaheuristic algorithms [7-11]. A metaheuristic is an iterative process with set of concepts that are used for exploring and exploiting the search space to determine the best solution among the alternative solutions. Metaheuristic algorithms are not problem specific, approximate and usually non-deterministic. It is important that there should be a dynamic balance between diversification and intensification in metaheuristic procedure. Diversification generally refers to the exploration of the search space and intensification refers to the exploitation of the accumulated search experience [7]. The balance between these two concepts is important so as not waste too much time in regions of the search space which does not possess high quality solutions while the algorithm can quickly find out the regions of high quality solutions. Some of the metaheuristic algorithms employ strategies that are inspired from nature. They simulate natural phenomena such as survival of the fittest, immune system, swarm intelligence and the cooling process of molten metals through annealing into a numerical algorithm. They are named according to the natural phenomena their search strategy is based such as evolutionary algorithms, immune system algorithm, particle swarm optimization and simulated annealing. Metaheuristic methods are non-traditional stochastic search and optimization methods and they are very suitable and efficient in finding the solution of combinatorial optimization problems.

It is shown in the literature that harmony search method which is one of the recently-developed metaheuristic techniques is quite effective and robust in solving structural optimization problems [12-19]. Performance evaluation of seven metaheuristic technique used in optimum design of pin jointed and rigidly jointed real size steel frames is carried in [16, 19]. In these studies it is shown that harmony search algorithm is quite successful stochastic search method and its performance in some problems is better than some other metaheuristic methods. Since its emergence, numbers of enhancements are suggested in order to improve the performance of the standard harmony search method. In this chapter these improvements are employed in solving the optimum design problem of steel space frames described above and their performances are compared.

6.3 Harmony Search Algorithms

The harmony search algorithm (HS) is originated by Geem et al. [2]. The algorithm was inspired by using the musical performance processes that occur when a musician searches for a perfect state of harmony, such as during jazz improvisation [2-6]. The analogy between finding a pleasing harmony in music and the optimum solution in an optimization problem is illustrated in Figure 6.2. A musician always intends to procedure a piece of music with perfect harmony. On the other hand, the optimal solution of an optimization problem should be the best solution available to the problem under given objective and limited by constraints. Both processes aim at reaching the best solution that is the optimum.

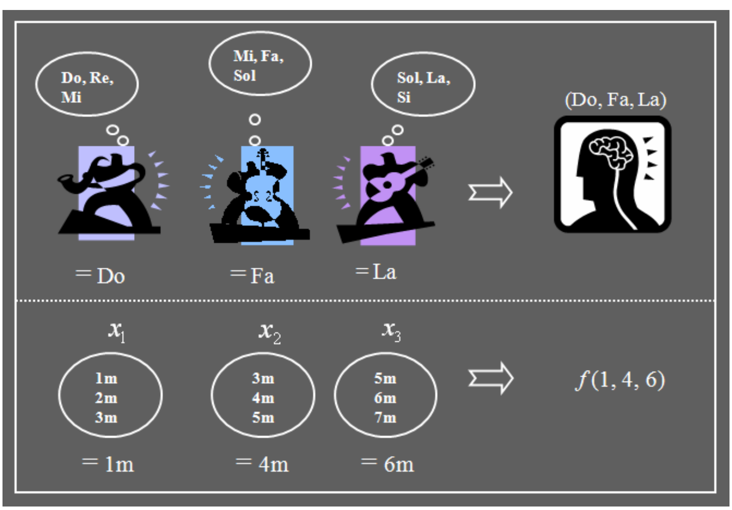


Fig. 6.2 Analogy between music improvisation and optimization [5]

6.3.1 Standard Harmony Search Algorithm

Harmony search method imitates the improvisation process of a skilled musician. When a musician is improvising, he or she has three possible choices: (a) can play any tune from his or her memory; (b) can play something similar to aforementioned tune by just adjusting pitch slightly; (c) can play a tune completely new. These three options are simulated in three components in harmony search method. These are usage of harmony memory matrix (**H**), pitch adjusting (*par*) and randomization. Before initiating the design process, a set of steel sections selected from an available profile list are collected in a design pool. Each steel section is assigned a sequence number I that varies between 1 to total number of sections (N_{sec}) in the list. It is important to note that during optimization process selection of sections for design variables is carried out using these numbers. The steps of the algorithm are outlined in the following as given in [2]:

6.3.1.1 Initialization of Harmony Memory Matrix

A harmony memory matrix $\mathbf{H}$ given in Eqn. (6.12) is randomly generated. The harmony memory matrix simply represents a design population for the solution of a problem under consideration, and incorporates a predefined number of solution vectors referred to as harmony memory size (hms). Each solution vector (harmony vector, $\mathbf{I}^i$) consists of ng design variables, and is represented in a separate row of the matrix; consequently the size of $\mathbf{H}$ is $(hms \times ng)$.

$$\mathbf{H} = \begin{bmatrix} I_1^1 & I_2^1 & \ldots & I_{ng}^1 \\ I_1^2 & I_2^2 & \ldots & I_{ng}^2 \\ \ldots & \ldots & \ldots & \ldots \\ I_1^{hms} & I_2^{hms} & \ldots & I_{ng}^{hms} \end{bmatrix} \begin{matrix} \phi(\mathbf{I}^1) \\ \phi(\mathbf{I}^2) \\ \ldots \\ \phi(\mathbf{I}^{hms}) \end{matrix} \tag{6.12}$$

6.3.1.2 Evaluation of Harmony Memory Matrix

(hms) solutions are then analyzed, and their objective function values are calculated. The solutions evaluated are sorted in the matrix in the increasing order of objective function values, that is $\phi(\mathbf{I}^1) \leq \phi(\mathbf{I}^2) \leq \ldots \leq \phi(\mathbf{I}^{hms})$.

6.3.1.3 Improvising a New Harmony

In harmony search algorithm the generation of a new solution (harmony) vector is controlled by two parameters ($hmcr$ and par) of the technique. The harmony memory considering rate ($hmcr$) refers to a probability value that biases the algorithm to select a value for a design variable either from harmony memory or from the entire set of discrete values used for the variable. That is to say, this parameter decides in what extent previously visited favorable solutions should be considered in comparison to exploration of new design regions while generating new solutions. At times when the variable is selected from harmony memory, it is checked whether this value should be substituted with its very lower or upper neighboring one in the discrete set. Here the goal is to encourage a more explorative search by allowing transitions to designs in the vicinity of the current solutions. This phenomenon is known as pitch-adjustment in harmony search method, and is controlled by pitch adjusting rate parameter (par). In the standard algorithm both of these parameters are set to suitable constant values for all harmony vectors generated regardless of whether an exploitative or explorative search is indeed required at a time during the search process. Accordingly a new harmony $\mathbf{I}' = \left[I_1', I_2',.., I_{ng}'\right]$ is improvised (generated) by selecting each design variable from either harmony memory or the entire discrete set. The probability that a design variable is selected from the harmony memory is controlled by harmony memory considering rate $(hmcr)$. To execute this probability, a random number r_i is generated between 0 and 1 for each variable I_i. If r_i is smaller than or equal to $hmcr$, the variable is chosen from harmony memory. Otherwise, a random value is assigned to the variable from the entire discrete set as shown in Eqn. (6.13).

$$I_i' = \begin{cases} I_i' \in \left\{I_i^1, I_i^2, ..., I_i^{hms}\right\} & if \;\; r_i \le hmcr \\ I_i' \in \left\{1, ..., ng\right\} & if \;\; r_i > hmcr \end{cases} \tag{6.13}$$

If a design variable attains its value from harmony memory, it is checked whether this value should be pitch-adjusted or not. In pitch adjustment, the value of a design variable is altered to its very upper or lower neighboring value obtained by adding ± 1 to its current value. Similar to $(hmcr)$ parameter, it is operated with a probability known as pitch adjustment rate (par). If not activated by (par), the value of the variable does not change as given in Eqn. (6.14).

$$I_i'' = \begin{cases} I_i' \pm bw & if \;\; r_i \le par \\ I_i' & if \;\; r_i > par \end{cases} \tag{6.14}$$

where, bw is arbitrary distance bandwidth which is taken as 1 in the standard harmony search method.

6.3.1.4 Update of Harmony Matrix

After generating the new harmony vector, its objective function value is calculated. If this value is better (lower) than that of the worst harmony vector in the harmony memory, it is then included in the matrix while the worst one is discarded out of the matrix. The updated harmony memory matrix is then sorted in ascending order of the objective function value.

6.3.1.5 Termination

The steps 3.1.2 and 3.1.3 are repeated until a pre-assigned maximum number of cycles are reached.

6.4 Various Harmony Search Algorithms

Within the recent years, number of enhancements is suggested to standard harmony search method in order to improve its performance. In this study, seven variations of harmony search algorithms are considered to determine the solution of the optimum design problem of steel space frames. These techniques are summarized in the following.

6.4.1 Standard Harmony Search with Adaptive Error Strategy (SHSAES)

This version is same as the standard harmony search method. It follows the same steps explained above. The only difference is that in addition to feasible solution vectors slightly infeasible solution vectors are also included in the harmony memory matrix. The candidate solution vectors that violate one or more design constraints slightly are also accepted as solutions due to the fact that they may possess some appropriate values for some of the design variables which can be used in pitch adjusting in the next iteration. It should be noticed that such candidate design vectors are allowed in the beginning phase of the design process but they are required to be taken out from the harmony memory matrix towards final phases of design cycles. This achieved by using larger error value initially and then this value is adjusted during the design cycles according to the expression given below.

$$Tol(i) = Tol_{\max} - \frac{(Tol_{\max} - Tol_{\min}) \cdot i^{0.5}}{(iter_{\max})^{0.5}} \tag{6.15}$$

where, $Tol(i)$ is the error value in iteration i, $Tol_{\max}$ and $Tol_{\min}$ are the maximum and the minimum error values defined in the algorithm respectively, $iter_{\max}$ is the maximum iteration number until which tolerance minimization procedure continues. Equation (6.15) provides larger error values in the beginning of the design cycles and quite small error values towards the final design cycles. Hence when the maximum design cycles are reached the acceptable design vectors remain in the harmony memory matrix and the ones which do not satisfy one or more design constraints smaller than the error tolerance would be pushed out during the design iterations.

6.4.2 Standard Harmony Search with Penalty Function (SHSPF)

In this application of the harmony search method also follows the steps of the standard harmony search technique. It only differs from the standard one in the acceptance of the candidate solution vectors. All the design vectors selected randomly are included in the harmony memory matrix regardless of whether they satisfy the design constraints in design problem or not. However, a penalty function is constructed as shown in the following.

$$W_p = W(1+C)^{\varepsilon} \tag{6.16}$$

where, W_p is the objective function that contain the penalty and W is the original objective function which is taken as the minimum weight of the steel space frames as given in Eqn. (6.1). C is the total constraint violation value calculated from the sum of the values of constraints function violations as given in equation (6.17). ε is the penalty coefficient taken as 2.

$$C = \sum C_s + \sum C_d + \sum C_{id} + \sum C_{td} + \sum C_{cd} + \sum C_{cc} + \sum C_{bc} \tag{6.17}$$

where, C_s, C_d, C_{id}, C_{td}, C_{cd}, C_{cc} and C_{bb} are the constraint functions violations for strength, deflection, inter-story drift, top story drift, column-to-column depth and unit weight and beam-to-column geometric constraint functions given in inequalities (6.2), (6.3), (6.5), (6.6), (6.7), (6.8), (6.9), (6.10) and (6.11) respectively. In general form, constraint function violation is calculated as:

$$C_i = \begin{cases} 0 & if \;\; g_i(x_j) \le 0 \qquad i = 1,\ldots nc \\ g_i(x_j) & if \;\; g_i(x_j) > 0 \qquad j = 1,\ldots ng \end{cases} \tag{6.18}$$

where, $g_i(x)$ is i[th] constraint function, x is the vector of design variables, nc is the total number of constraint functions and ng is the total number of member groups (the total number of design variables) in the optimum design problem. It is apparent from Eqn. (6.18) that feasible solutions will not be subjected to any penalty and their objective function value will be equal to the original objective function value given in Eqn. (6.1). The harmony search method seeks solution vectors in the design space that have smaller objective function values and stores these in the harmony memory matrix during the design cycles. As a result those solution vectors that have larger objective function values are eliminated from the harmony memory matrix within the harmony search iterations. Towards the end of design cycles only those solution vectors that do not have any penalty remains in the harmony memory matrix and among these that have the least weight represents the optimum solution.

6.4.3 Adaptive Harmony Search with Penalty Function (AHSPF)

In standard harmony search method there are two parameters known as harmony memory considering rate (*hmcr*) and pitch adjusting rate (*par*) that play an important role in obtaining the optimum solution. These parameters are assigned to constant values that are arbitrarily chosen within their recommended ranges by Geem [2-6] based on the observed efficiency of the technique in different problem fields. It is observed through the application of the standard harmony search method that the selection of these values is problem dependent. While a certain set of values yields a good performance of the technique in one type of design problem, the same set may not present the same performance in another type of design problem. Hence it is not possible to come up with a set of values that can be used in every optimum design problem. In each problem a sensitivity analysis is required to be carried out determine what set of values results a good performance. Adaptive harmony search method eliminates the necessity of finding the best set of parameter values [22, 23]. It adjusts the values of these parameters automatically during the optimization process. The basic components of the adaptive harmony search algorithm are outlined as follows.

6.4.3.1 Initialization of a Parameter Set

Harmony search method uses four parameters values of which are required to be selected by the user. This parameter set consists of a harmony memory size (*hms*), a harmony memory considering rate (*hmcr*), a pitch adjusting rate (*par*) and a maximum search number ($N_{\max}$). Out of these four parameters, (*hmcr*) and (*par*) are made dynamic parameters in adaptive harmony search method that vary from one solution vector to another. They are set to initial values of $hmcr^{(0)}$ and $par^{(0)}$ for all the solution vectors in the initial harmony memory matrix. In the standard harmony search algorithm these parameters are treated as static quantities, and they are assigned to suitable values chosen within their recommended ranges of $hmcr \in$ [0.70, 0.95] and $par \in$ [0.20, 0.50] [2-6].

6.4.3.2 Initialization and Evaluation of Harmony Memory Matrix

The harmony memory matrix is established randomly as explained in section 3.1.1 which contains candidate design vectors for the optimum design problem under consideration. The structural analysis of each solution is then performed with the set of steel sections selected for design variables, and responses of each candidate solution are obtained under the applied loads. The objective function values of the feasible solutions that satisfy all problem constraints are directly calculated from Eqn. (6.1). However, infeasible solutions that violate some of the problem constraints are penalized using external penalty function approach, and their objective function values are calculated according to Eqn. (6.19).

$$\phi = W\left[1+\alpha\left(\sum_i g_i\right)\right] \tag{6.19}$$

In Eqn. (6.19), ϕ is the constrained objective function value, g_i is the i-th problem constraint value and α is the penalty coefficient used to tune the intensity of penalization as a whole. This parameter is set to an appropriate static value of $\alpha = 1$ in the numerical examples. Finally, the solutions evaluated are sorted in the matrix in the descending order of objective function values, that is, $\phi(\mathbf{I}^1) \leq \phi(\mathbf{I}^2) \leq \ldots \leq \phi(\mathbf{I}^{hms})$.

6.4.3.3 Generating a New Harmony Vector

In the adaptive algorithm a new set of values is sampled for $hmcr$ and par parameters each time prior to improvisation (generation) of a new harmony vector, which in fact forms the basis for the algorithm to gain adaptation to varying features of the design space. Accordingly, to generate a new harmony vector in the algorithm proposed, a two-step procedure is followed consisting of (i) sampling of control parameters, and (ii) improvisation of the design vector.

6.4.3.3.1 Sampling of Control Parameters

For each harmony vector to be generated during the search process, first a new set of values are sampled for $hmcr$ and par control parameters by applying a logistic normal distribution based variation to the average values of these parameters within the harmony memory matrix, as formulated in Eqns. (6.20 and 6.21).

$$(hmcr)^k = \left(1+\frac{1-(hmcr)'}{(hmcr)'}.e^{-\gamma.N(0,1)}\right)^{-1} \tag{6.20}$$

$$(par)^k = \left(1+\frac{1-(par)'}{(par)'}.e^{-\gamma.N(0,1)}\right)^{-1} \tag{6.21}$$

In Eqns. (6.20) and (6.21), $(hmcr)^k$ and $(par)^k$ represent the sampled values of the control parameters for a new harmony vector. The notation $N(0,1)$ designates a normally distributed random number having expectation 0 and standard deviation 1. The symbols $(hmcr)'$ and $(par)'$ denote the average values of control parameters within the harmony memory matrix, obtained by averaging the corresponding values of all the solution vectors within the $\mathbf{H}$ matrix, that is,

$$(hmcr)' = \frac{\sum_{i=1}^{hms} (hmcr)^i}{(hms)} \quad , \quad (par) = \frac{\sum_{i=1}^{hms} (par)^i}{(hms)} \tag{6.22}$$

Finally, the factor γ in Eqns. (6.20) and (6.21) refers to the learning rate of control parameters, which is recommended to be selected within a range of [0.25, 0.50]. In the numerical examples this parameter is set to 0.35.

In this implementation, for each new vector a probabilistic sampling of control parameters is motivated around average values of these parameters $(hmcr)'$ and $(par)'$ observed in the $\mathbf{H}$ matrix. Considering the fact that the harmony memory matrix at an instant incorporates the best (hms) solutions sampled thus far during the search, to encourage forthcoming vectors to be sampled with values that the search process has taken the most advantage in the past. The use of a logistic normal distribution provides an ideal platform in this sense because not only it guarantees the sampled values of control parameters to lie within their possible range of variation, i.e., [0, 1], but also it permits occurrence of small variations around $(hmcr)'$ and $(par)'$ more frequently than large ones. Accordingly, sampled values of control parameters mostly fall within close vicinity of the average values, yet remote values are occasionally promoted to check alternating demands of the search process.

6.4.3.3.2 Improvisation of the Design Vector

Upon sampling of a new set of values for control parameters, the new harmony vector $\mathbf{I}^k = \left[I_1^k, I_2^k, .., I_{ng}^k\right]$ is improvised in such a way that each design variable is selected at random from either harmony memory matrix or the entire discrete set. Which one of these two sets is used for a variable is determined probabilistically in conjunction with harmony memory considering rate $(hmcr)^k$ parameter of the solution. To implement the process a uniform random number r_i is generated between 0 and 1 for each variable I_i^k. If r_i is smaller than or equal to $(hmcr)^k$, the variable is chosen from harmony memory in which case it is assigned any value from the i-th column of the $\mathbf{H}$ matrix, representing the value set of the variable in (hms) solutions of the matrix (Eqn. 6.12). Otherwise (if $r_i > (hmcr)^k$), an arbitrary value is assigned to the variable from the entire design set.

$$I_i^k = \begin{cases} I_i^k \in \left\{I_i^1, I_i^2, ..., I_i^{hms}\right\} & if \;\; r_i \le (hmcr)^k \\ I_i^k \in \{1, ..., N_{sec}\} & if \;\; r_i > (hmcr)^k \end{cases} \tag{6.23}$$

If a design variable attains its value from harmony memory, it is checked whether this value should be pitch-adjusted or not. In pitch adjustment the value of a design variable ($I_i^{k'}$) is altered to its very upper or lower neighboring value obtained by adding ± 1 to its current value. This process is also operated probabilistically in conjunction with pitch adjusting rate $(par)^k$ parameter of the solution, Eqn. (6.21). If not activated by $(par)^k$, the value of the variable does not change. Pitch adjustment prevents stagnation and improves the harmony memory for diversity with a greater chance of reaching the global optimum.

$$I_i^{k'} = \begin{cases} I_i^k \pm 1 & \text{if } r_i \le (par)^k \\ I_i^k & \text{if } r_i > (par)^k \end{cases} \tag{6.24}$$

6.4.3.4 Update of the Harmony Memory and Adaptivity

After generating the new harmony vector, its objective function value is calculated as per Eqn. (6.19). If this value is better (lower) than that of the worst solution in the harmony memory matrix, it is included in the matrix while the worst one is discarded out of the matrix. It follows that the solutions in the harmony memory matrix represent the best (hms) design points located thus far during the search. The harmony memory matrix is then sorted in ascending order of objective function value. Whenever a new solution is added into the harmony memory matrix, the $(hmcr)'$ and $(par)'$ parameters are recalculated using Eqn. (6.22). This way the harmony memory matrix is updated with the most recent information required for an efficient search and the forthcoming solution vectors are guided to make their own selection of control parameters mostly around these updated values. It should be underlined that there are no single values of control parameters that lead to the most efficient search of the algorithm throughout the process unless the design domain is completely uniform. On the contrary, the optimum values of control parameters have a tendency to change over time depending on various regions of the design space in which the search is carried out. The update of the control parameters within the harmony memory matrix enables the algorithm to catch up with the varying needs of the search process as well. Hence the most advantages values of control parameters are adapted in the course of time automatically (i.e., by the algorithm itself), which plays the major role in the success of *adaptive* harmony search method discussed in the paper.

6.4.3.5 Termination

The steps 4.3.4 and 4.3.5 are iterated in the same manner for each solution sampled in the process, and the algorithm terminates when a predefined number of solutions ($N_{\max}$) is sampled.

6.4.4 Improved Harmony Search (IHS, Mahdavi)

Standard harmony search method uses fixed values for both pitch adjustment rate *(par)* and arbitrary distance width *(bw)*. Prior to the application of the algorithm some appropriate values are selected for these parameters and they are kept the same until the end of the iterations. For example the value of the arbitrary distance bandwidth *(bw)* is taken as ±1 in the standard harmony search method, although some other value can also be used if preferred. It is also stated in the work of Mahdavi et. al. [20] that the use of for example small fixed values for pitch adjustment rate *(par)* with large values of arbitrary distance bandwidth *(bw)* can cause considerable increase in the total number of iterations required to reach the optimum solution, resulting in an undesirable poor performance of the algorithm. In order to avoid such a poor performance of the algorithm they have suggested adaptive expressions for both of these parameters instead of fixed values. The values of these parameters are adjusted dynamically by using Equations (6.25) and (6.26) during the harmony search iterations. However, the fixed value is used for the harmony memory consideration rate *(hmcr)* which is kept the same until the end of the iterations as in the standard harmony search method.

$$par(i) = par_{\min} + \frac{(par_{\max} - par_{\min})}{Iter_{\max}} \times i \qquad i = 1,2,\ldots, Iter_{\max} \tag{6.25}$$

where; $par(i)$ is pitch adjusting rate at iteration i, $par_{\max}$ and $par_{\min}$ are the maximum and the minimum values of pitch adjusting rates, $Iter_{\max}$ is the maximum iteration number.

$$bw(i) = bw_{\max} \exp\left(\frac{\ell n\left(\frac{bw_{\min}}{bw_{\max}}\right)}{Iter_{\max}} \times i \right) \tag{6.26}$$

where; $bw(i)$ is bandwidth in iteration i, $bw_{\max}$ and $bw_{\min}$ are the maximum and the minimum distance bandwidth. $par_{\max}$, $par_{\min}$, $bw_{\max}$ and $bw_{\min}$ are specified prior to the application of the algorithm. They are taken as 0.5, 0.05, 1 and 5 respectively. In this study this technique is used with adaptive error strategy explained in section 4.1 not with penalty function concept.

6.4.5 Global Best Harmony Search (GBHS, Mahdavi)

Mahdavi [5] also suggested another enhancement which makes use of the concept of particle swarm optimizer [29, 30] to the improved harmony search method [20]. In particle swarm optimizer system, a swarm of particles fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by its best position and also the position of the best particle in the swarm. The global best harmony search modifies the pitch adjustment step of the harmony search method similar to particle swarm optimizer. It replaces the arbitrary distance width *(bw)* altogether and adds a social dimension to the harmony search by selecting the best harmony for the new harmony vector. The algorithm computes *(par)* from the equation (6.25) and but does not use *(bw)* at all. Instead, it employs the following equation to construct the new harmony vector, not the one given in Eq. (6.14).

$$I_i^{new} \to I_k^{best} \tag{6.27}$$

where; *best* is the index of the best harmony in the harmony memory matrix and *k* is the variable number randomly selected between 1 to *ng* which is the total number of design variables in the optimum design problem. In this study this technique is used with adaptive error strategy explained in section 4.1 not with penalty function concept.

6.4.6 Improved Harmony Search (IHSC, Coelho)

In another enhancement to standard harmony search algorithm Coelho et al. [24] has suggested another adaptive expression given in Eqn. (6.28) for the pitch adjustment rate *(par)*. This version of harmony search method is also called improved harmony search method and it has the same steps of the standard harmony search algorithm with the exception that it changes the value of the pitch adjustment rate *(par)* each iteration with the value computed from the following equation.

$$par(i) = par_{\min} + \left(par_{\max} - par_{\min}\right) \times \frac{\left(HMVal_{\max}(i) - Mean\,(HMVal)\right)}{\left(HMVal_{\max}(i) - HMVal_{\min}(i)\right)} \tag{6.28}$$

where; $par(i)$ is the value of the pitch adjusting rate in iteration i, $par_{\max}$ and $par_{\min}$ are the maximum and the minimum values of pitch adjusting rates respectively, $HMVal_{\max}$, $HMVal_{\min}$ and $Mean\,(HMVal)$ are the minimum, the maximum and the mean values of objective function in the harmony memory matrix respectively. The values of $par_{\max}$ and $par_{\min}$ are taken as 0.99 and 0.01 respectively in this study. In this study this technique is used with adaptive error strategy explained in section 4.1 not with penalty function concept.

6.4.7 Dynamic Harmony Search (DHS)

Dynamic harmony search is suggested in this study. This version of the harmony search method has the same steps of the standard harmony search algorithm with the exception that instead of using fixed values for both parameters of harmony memory considering rate *(hmcr)* and pitch adjusting rate *(par)* their values are calculated by means of adaptive expressions. The value of *(hmcr)* is computed from Eqn. (6.20) and *(par)* is calculated from Eqn. (6.28). In other words dynamic harmony search method is a mixture of adaptive harmony and Coelho's improved harmony search algorithms. The adaptive error strategy explained in section 4.1 but not the penalty function concept is employed in this technique as well.

6.5 Design Examples

Seven different structural optimization programs are coded each of which is based on one of the above explained versions of the harmony search algorithms. Three steel space frames are designed using these seven different versions of harmony search algorithms and the optimum solutions determined are compared with each other in order to evaluate the performance of each version.

6.5.1 Five-Story, Two-Bay Regular Steel Space Frame

The plan and 3D views of the five-story, two-bay steel frame shown in the Figures 6.3 and 6.4 is a regular steel frame with 54 joints and 105 members that are grouped into 11 independent design variables. The frame is subjected to gravity loads as well as lateral loads that are computed as per ASCE 7-05 [28]. The design dead and live loads are taken as 2.88kN/m^2 and 2.39kN/m^2 respectively. The ground snow load is considered to be 0.755kN/m^2 and a basic wind speed is 105mph (65 m/s). The un-factored distributed gravity loads on the beams of the roof and floors are tabulated in Table 6.2. The following load combinations are considered in the design of the frame according to the code specification. 1.2D+1.6L+0.5S, 1.2D+0.5L+1.6S, 1.2D+1.6W+0.5L+0.5S where D is the dead load, L represents the live load, S is the snow load and W is the wind load. The drift ratio limits of this frame are defined as 1.33 cm for inter story drift and 6.67 cm for top story drift. Maximum deflection of beam members is restricted as 1.67 cm.

Table 6.2 Beam gravity loading of the five-story, two bay steel frame

Beam Type	Uniformly distributed load (kN/m)		
	Dead Load	Live Load	Snow Load
Roof Beams	4.78	-	1.508
Floor Beams	4.78	5.76	-

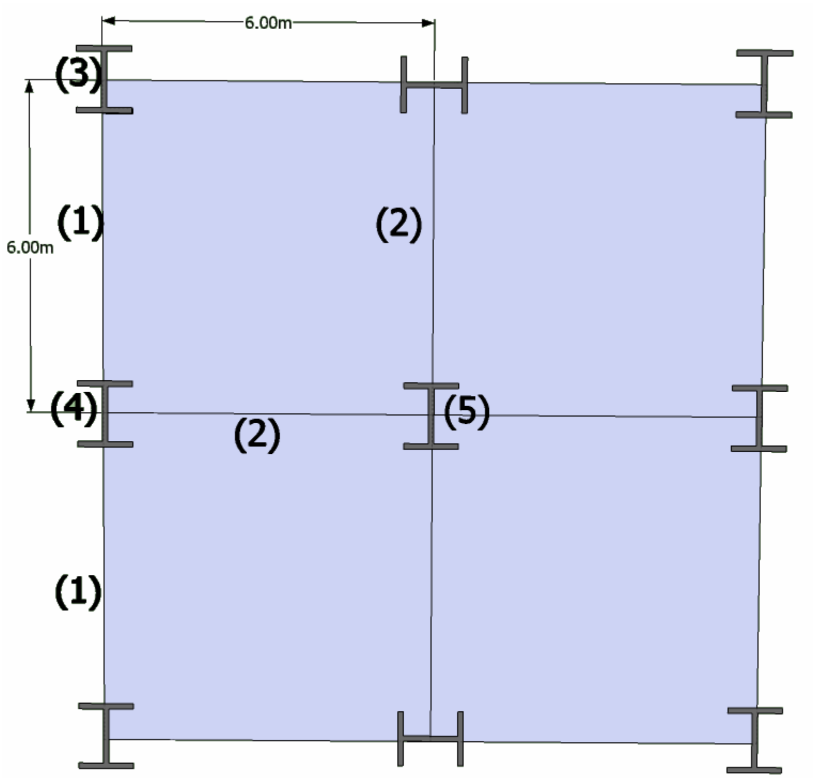


Fig. 6.3 Plan view of five-story, two bay steel frame

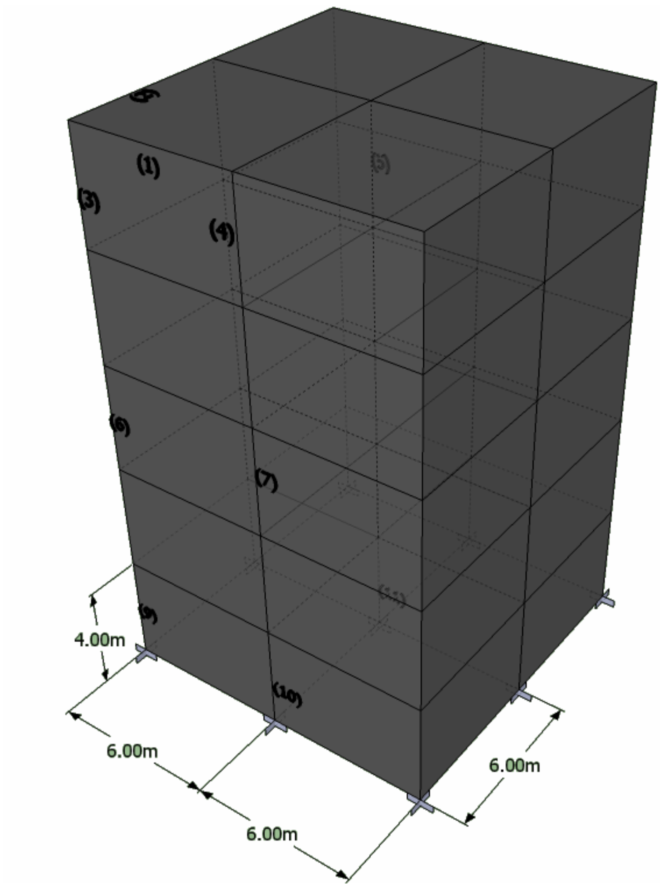


Fig. 6.4 3D View of the five-story, two bay steel frame

Optimum design problem of the five-story, two-bay steel frame is solved by using seven different versions of harmony search algorithms. In these algorithms the following harmony search parameters are used: harmony memory size (hms) = 20, pitch adjusting rate (par) = 0.3, harmony memory considering rate (hmcr) = 0.9 and maximum iteration number = 50000. The optimum designs obtained from each of these algorithms are shown in Table 6.3. It is apparent from the table that the lightest weight is 261.128 kN which is obtained by the adaptive harmony search algorithm and the second lightest design is 261.360kN attained by the dynamic harmony search method suggested in this study. The design histories of these algorithms for the best solutions are plotted in Fig. 6.5. It is apparent from the figure that the dynamic harmony search and adaptive harmony search algorithms show better performance than others. It is noticed that the minimum weight determined by the dynamic harmony search and adaptive harmony search algorithms are 12.3% less than the heaviest frame.

Table 6.3 Design results of the five-story, two bay steel frame

Member Group	Type	SHSAES	SHSPF	AHSPF
1	Beam	W530X66	W360X39	W410X46.1
2	Beam	W310X38.7	W310X38.7	W310X38.7
3	Column	W200X35.9	W360X39	W250X32.7
4	Column	W200X35.9	W200X46.1	W200X46.1
5	Column	W360X44	W610X101	W460X52
6	Column	W310X38.7	W530X66	W410X53
7	Column	W360X72	W410X60	W360X64
8	Column	W610X92	W1000X222	W920X201
9	Column	W410X53	W610X92	W410X53
10	Column	W360X72	W410X60	W460X74
11	Column	W760X147	W1100X390	W1000X258
Max. Strength Ratio		0.979	0.986	0.936
Top Drift(cm)		4.837	5.264	4.81
Inter Story Drift(cm)		1.333	1.329	1.331
Maximum Iteration		50000	50000	50000
Weight (kN)		278.196	268.172	261.128

Member Group	Type	IHS (Mahdavi)	GBHS (Mahdavi)	IHSC (Coelho)	DHS Present Study
1	Beam	W530X66	W530X74	W530X74	W460X52
2	Beam	W310X38.7	W360X44	W360X44	W250X38.5
3	Column	W200X35.9	W200X41.7	W200X41.7	W310X38.7
4	Column	W200X35.9	W200X41.7	W200X41.7	W200X35.9
5	Column	W360X44	W360X44	W360X44	W460X52
6	Column	W310X38.7	W360X44	W360X44	W360X51
7	Column	W360X64	W360X44	W310X44.5	W250X73
8	Column	W610X82	W610X92	W610X113	W610X101
9	Column	W410X53	W360X51	W360X51	W360X51
10	Column	W360X64	W410X60	W410X60	W460X74
11	Column	W840X193	W840X193	W760X134	W840X193
Max. Strength Ratio		0.989	0.986	0.979	0.975
Top Drift(cm)		4.763	4.579	4.73	5.074
Inter Story Drift(cm)		1.331	1.325	1.333	1.33
Maximum Iteration		50000	50000	50000	50000
Weight (kN)		275.46	297.928	297.424	261.36

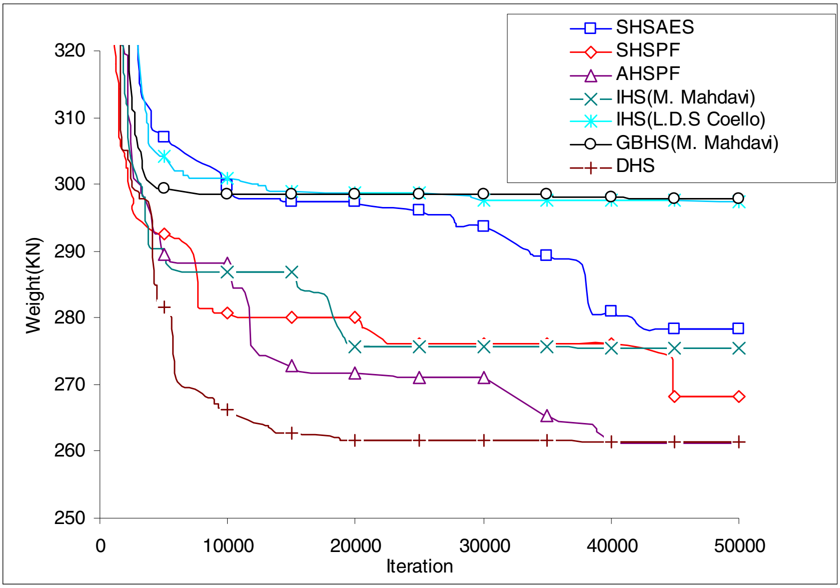


Fig. 6.5 Design histories of the five-story, two bay steel frame

6.5.2 *Ten-Story, Four-Bay Steel Space Frame*

The three dimensional view, side view and the plan of ten-story four-bay steel frame shown in Figures 6.6 and 6.7 is taken from [19, 22]. This frame has 220 joints and 568 members which are collected in 25 member groups which are the independent design variables as shown in Figure 6.6. Inner roof beams, outer roof beams, inner floor beams and outer floor beams are subjected to 14.72kN/m, 7.36kN/m, 21.43kN/m and 10.72kN/m uniformly distributed gravity loads respectively. Lateral forces acting at the level of each story of the steel space frame are tabulated in Table 6.3. Drift ratio limits are defined as $h/400$ where h is the story height for inter story drift and $H/400$ for top story drift where H is the total height of the structure.

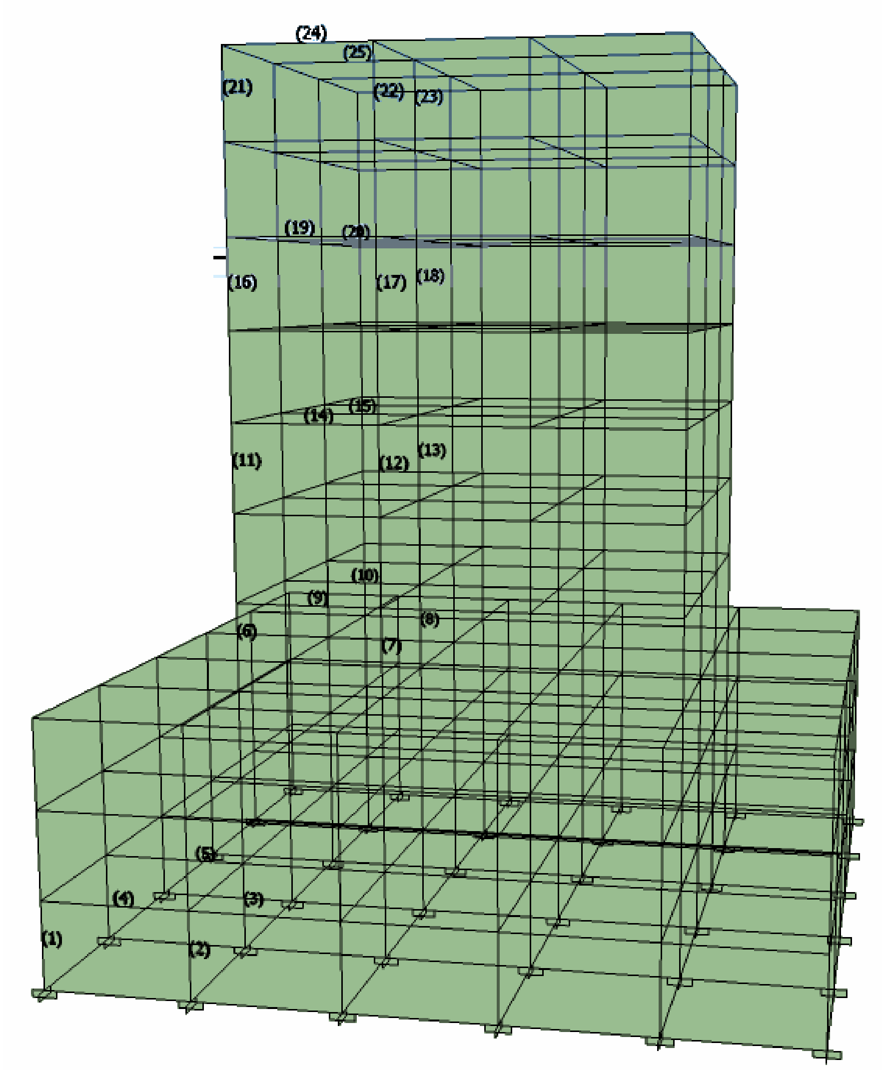


Fig. 6.6 3-D view of ten-story, four-bay steel space frame

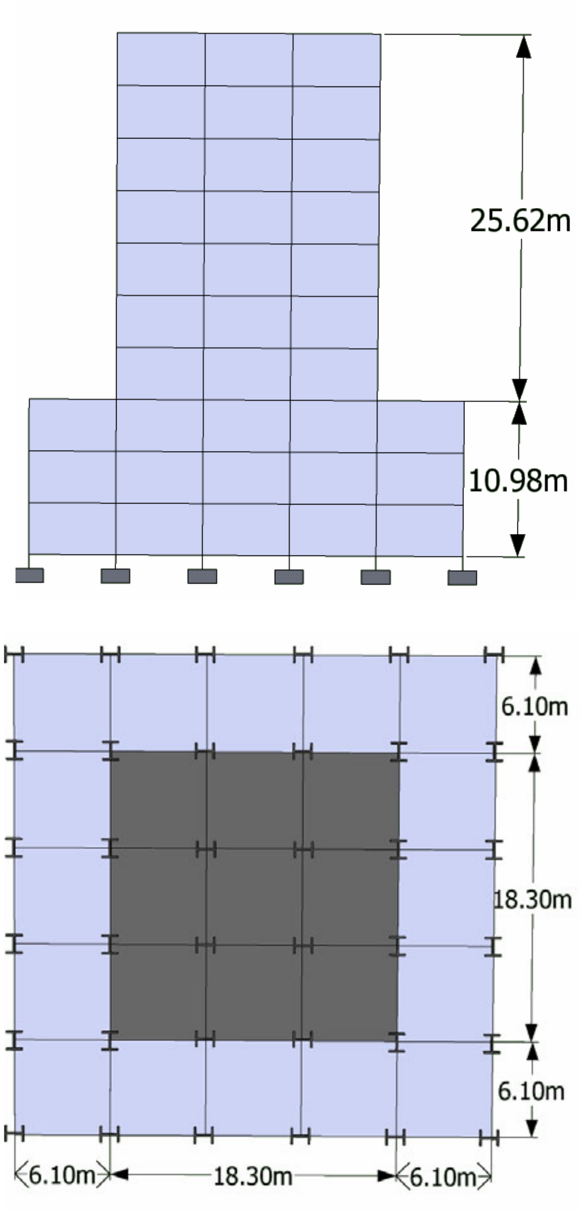


Fig. 6.7 Plan and side view of ten-story, four-bay steel space frame

Table 6.4 Lateral loads acting at the level of each story of ten-story, four-bay steel space frame

Story Number	Windward		Leeward	
	(lb/ft)	(kN/m)	(lb/ft)	(kN/m)
1	12.51	0.1825	127.38	1.8585
2	28.68	0.4184	127.38	1.8585
3	44.68	0.6519	127.38	1.8585
4	156.86	2.2886	127.38	1.8585
5	167.19	2.4393	127.38	1.8585
6	176.13	2.5698	127.38	1.8585
7	184.06	2.6854	127.38	1.8585
8	191.21	2.7897	127.38	1.8585
9	197.76	2.8853	127.38	1.8585
10	101.9	1.5743	127.38	1.8585

Optimum design problem of this frame is solved under the design constraints described in section 2 by using seven different versions of harmony search algorithms described. In these algorithms the following harmony search parameters are used: harmony memory size (hms) = 30, pitch adjusting rate (par) = 0.3, harmony memory considering rate (hmcr) = 0.9, and maximum iteration number = 80000. The optimum designs obtained by each of these algorithms are shown in Table 6.5. It is clear from the table that the lightest weight is 1699.88kN which is obtained by the dynamic harmony search method and the second lightest weight of the frame is 1714.46kN attained by the adaptive harmony search algorithm. The design histories of these algorithms for the best solutions are plotted in Fig. 6.8. It is apparent from the figure that the dynamic harmony search algorithms shows steady convergence and outperforms others. It is noticed that the minimum weight determined by the dynamic harmony search is 7.3% less than the heaviest frame.

Table 6.5 Design results for ten-story, four-bay steel space frame

Member Group	Type	SHSAES	SHSPF	AHSPF
1	Column	W310X28.3	W150X22.5	W310X28.3
2	Column	W310X28.3	W200X86	W310X28.3
3	Column	W360X32.9	W760X173	W360X39
4	Beam	W410X46.1	W250X25.3	W410X46.1
5	Beam	W410X46.1	W410X38.8	W460X52
6	Column	W410X46.1	W410X114	W410X38.8
7	Column	W460X52	W760X196	W410X38.8
8	Column	W530X66	W840X176	W610X82
9	Beam	W310X23.8	W360X110	W310X23.8
10	Beam	W460X60	W690X152	W460X52
11	Column	W250X67	W410X100	W200X35.9
12	Column	W250X73	W460X128	W250X80
13	Column	W310X44.5	W690X170	W360X44
14	Beam	W310X97	W310X60	W460X113
15	Beam	W460X128	W530X85	W460X113
16	Column	W530X85	W310X97	W530X85
17	Column	W310X107	W310X117	W460X128
18	Column	W530X150	W530X85	W610X217
19	Beam	W690X170	W250X32.7	W760X173
20	Beam	W310X117	W410X46.1	W530X150
21	Column	W760X196	W310X97	W690X217
22	Column	W840X176	W200X59	W760X173
23	Column	W150X29.8	W410X60	W150X24
24	Beam	W250X73	W250X32.7	W250X49.1
25	Beam	W410X132	W310X38.7	W360X134
Max. Strength Ratio		0.99	1	0.995
Top Drift(cm)		8.158	7.639	7.695
Inter Story Drift(cm)		0.914	0.914	0.914
Maximum Iteration		80000	80000	80000
Weight (kN)		1756.56	1800.28	1714.46

Table 6.5 (*continued*)

Member Group	Type	IHS (Mahdavi)	GBHS (Mahdavi)	IHSC (Coelho)	DHS Present Study
1	Column	W310X28.3	W310X23.8	W200X19.3	W310X23.8
2	Column	W310X28.3	W360X32.9	W530X85	W310X28.3
3	Column	W360X39	W460X52	W410X132	W410X46.1
4	Beam	W410X38.8	W310X38.7	W310X23.8	W410X46.1
5	Beam	W530X66	W360X64	W460X60	W410X46.1
6	Column	W530X66	W530X150	W310X107	W410X46.1
7	Column	W410X46.1	W310X32.7	W760X196	W530X66
8	Column	W410X38.8	W690X125	W760X257	W530X66
9	Beam	W310X23.8	W310X23.8	W530X109	W310X23.8
10	Beam	W460X60	W360X32.9	W360X134	W460X52
11	Column	W250X67	W250X73	W310X97	W250X73
12	Column	W250X73	W250X58	W250X101	W250X67
13	Column	W360X44	W360X51	W690X170	W360X44
14	Beam	W310X97	W250X80	W410X60	W310X97
15	Beam	W410X100	W610X113	W530X85	W460X113
16	Column	W530X85	W530X85	W250X73	W610X92
17	Column	W310X107	W610X101	W250X101	W360X134
18	Column	W530X150	W690X140	W530X92	W460X128
19	Beam	W690X170	W610X174	W410X38.8	W840X176
20	Beam	W360X162	W610X155	W410X46.1	W360X134
21	Column	W760X196	W920X201	W250X73	W760X196
22	Column	W690X170	W1000X296	W200X59	W840X176
23	Column	W150X29.8	W150X24	W410X60	W150X24
24	Beam	W410X53	W250X49.1	W310X28.26	W250X49.1
25	Beam	W310X129	W840X193	W310X28.3	W530X150
Max. Strength Ratio		0.988	0.961	0.965	0.976
Top Drift(cm)		7.774	8.077	7.589	8.069
Inter Story Drift(cm)		0.915	0.914	0.913	0.912
Maximum Iteration		80000	80000	80000	80000
Weight (kN)		1739.47	1842.95	1773.51	1699.88

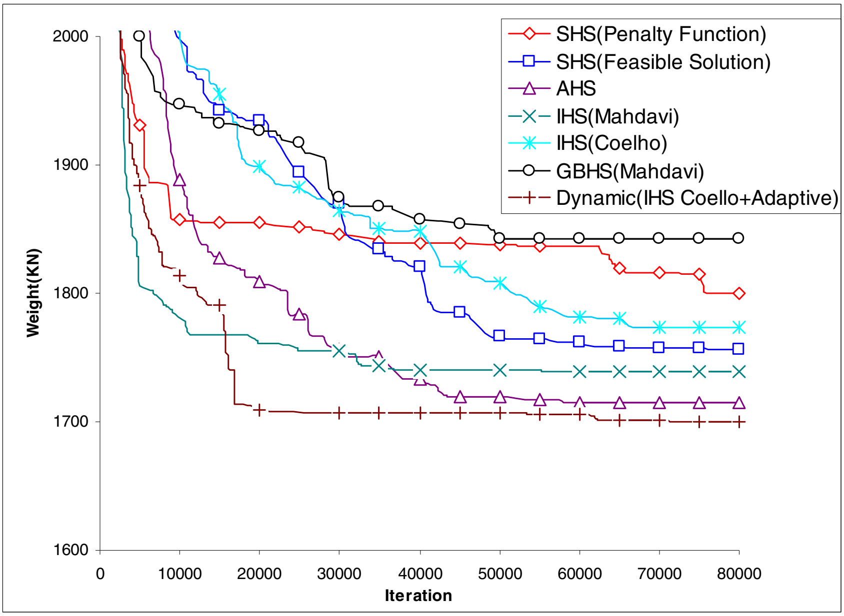
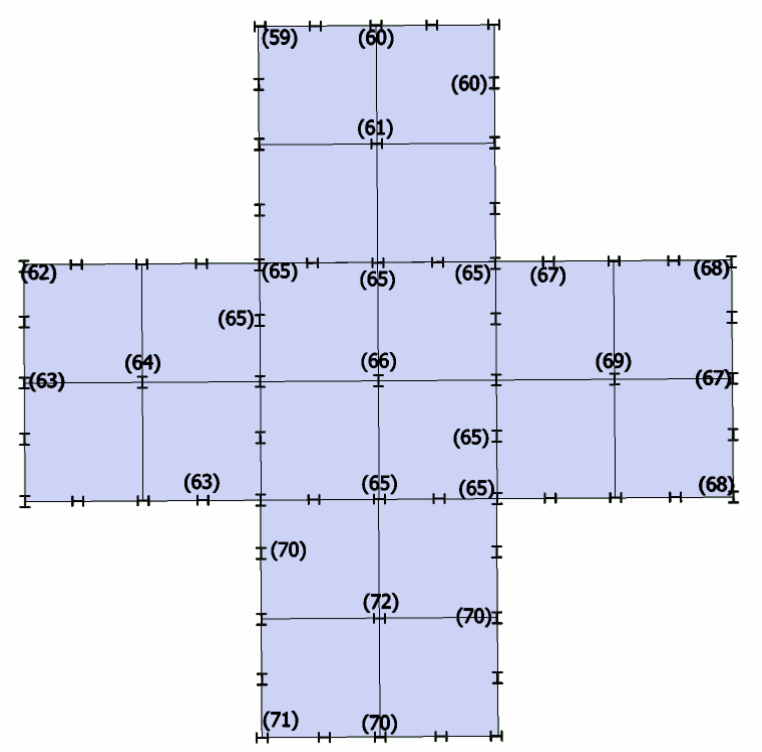


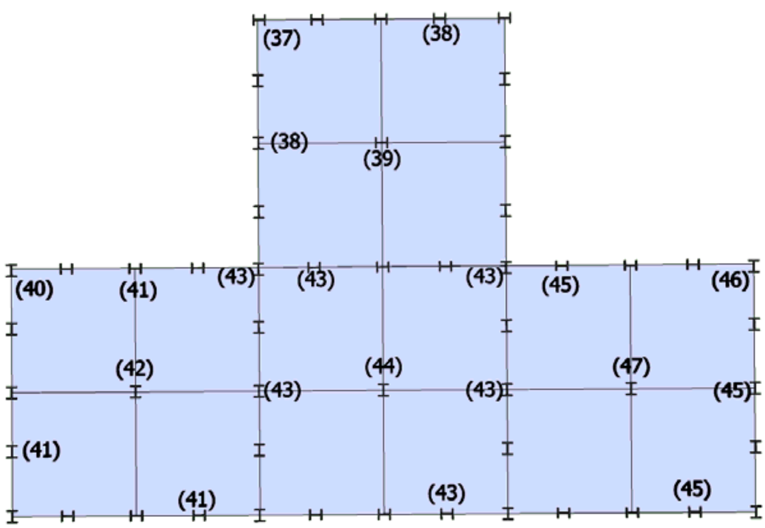
Fig. 6.8 Design histories of ten-story, four-bay steel space frame

6.5.3 Twenty-Story, 1860–Member, Steel Space Frame

The three dimensional view and plan of twenty-story, 1860-member steel space frame are illustrated in Figures 6.9 and 6.10. The frame has 820 joints and 1860 members which are collected in 86 independent design variables. The member grouping is given in Figure 6.9. The frame is subjected to gravity loads as well as lateral loads that are computed according to ASCE 7-05 [28]. The design dead and live loads are taken as 2.88kN/m^2 and 2.39kN/m^2 respectively. Basic wind speed is considered as 85mph (38 m/s). The following load combinations are considered in the design of the frame according to code specification [25]. 1.2D+1.3WZ+0.5L+0.5S and 1.2D+1.3WX+0.5L+0.5S where D is the dead load, L represents the live load, S is the snow load and WX, WZ are the wind loads in the global X and Z axis respectively. Drift ratio limits are defined as $h/400$ where h is the story height for inter story drift and $H/400$ for top story drift where H is the height of structure.

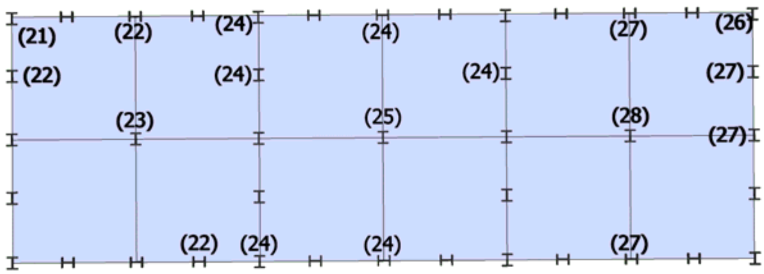


(a) Plan view of 1-4th story

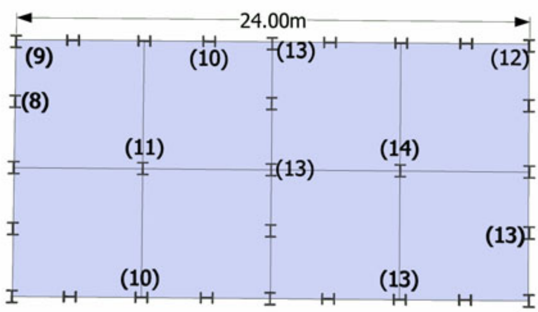


(b) Plan view of 5-8th story

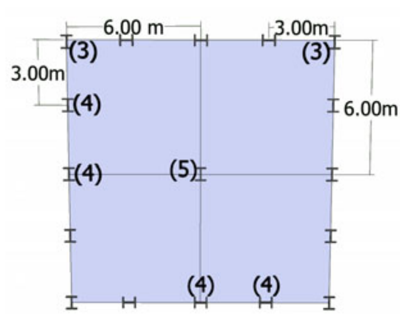
Fig. 6.9 Plan view of twenty-story, 1860 member steel space frame



(c) Plan view of 5-8th story



(d) Plan view of 13-16th story



(e) Plan view of 17-20th story

Fig. 6.9 (*continued*)

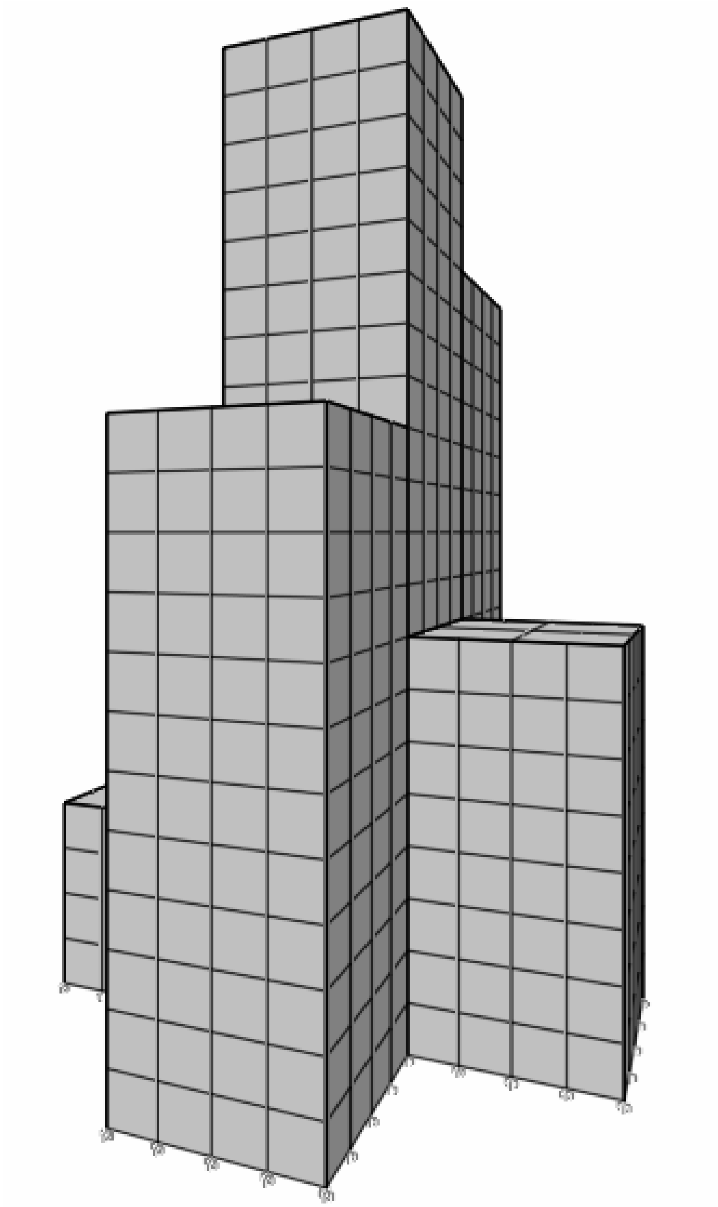


Fig. 6.10 3-D view of twenty-story, 1860 member steel space frame

This frame which has 1860 members is also designed seven times using different versions of harmony search algorithms. In these runs the harmony search parameters are selected as: harmony memory size (hms) = 50, the pitch adjusting rate (par) = 0.3, the harmony memory considering rate (hmcr) = 0.9, the maximum iteration number = 80000. The optimum designs obtained by each of these algorithms are given in Table 6.6. It is apparent from the table that the best design is obtained by dynamic harmony search method which has the minimum weight of 4716.576kN. The second best design is obtained by the adaptive harmony search algorithm (AHS) as 4932.012kN. Difference between these results is only 4%. However the minimum weights of best designs obtained by other harmony search algorithms are around 6000kN. Therefore, it can be stated that the dynamic and adaptive harmony search methods demonstrated better performance than the other versions of harmony search methods. The design histories of each harmony search method are shown in Fig. 6.11. The figure clearly reveals the fact that the dynamic and adaptive harmony search methods perform better than the other versions of the harmony search algorithms from the beginning of the design cycles.

Table 6.6 Design results for twenty-story, 1860-member steel space frame

Beam Type	Member Group	SHSAES	SHSPF	AHSPF
Outer	1	W410X67	W410X53	W460X60
Interior	2	W460X52	W460X68	W460X60
Columns Story	Member Group	SHSAES	SHSPF	AHSPF
20,19	3	W410X85	W250X73	W310X38.7
19,18	6	W410X132	W690X125	W410X38.8
16,15	9	W410X60	W200X41.7	W200X22.5
14,13	15	W920X223	W460X68	W200X26.6
12,11	21	W920X271	W610X174	W360X39
10,9	29	W1000X314	W840X176	W460X60
8,7	37	W150X29.8	W200X22.5	W250X25.3
6,5	48	W410X46.1	W460X128	W310X28.3
4,3	59	W1000X272	W840X176	W360X51
4,3	72	W310X44.5	W200X86	W250X32.7
2,1	73	W1000X272	W840X251	W690X125
Top Story Drift (cm)		9.013	8.777	9.809
Inter-Story Drift (cm)		0.742	0.738	0.75
Max. Strength Ratio		0.84	0.837	1
Weight (kN)		6319.554	6204.204	4932.012

Table 6.6 (*continued*)

Beam Type	Member Group	HIS (Mahdavi)	GBHS (Mahdavi)	HIS (Coelho)	DHS Present Study
Outer	1	W410X60	W410X60	W360X51	W410X53
Interior	2	W460X60	W410X60	W460X60	W460X60
Columns Story	**Member Group**	**HIS (Mahdavi)**	**GBHS (Mahdavi)**	**HIS (Coelho)**	**DHS (Present Study)**
20,19	3	W200X46.1	W250X80	W360X51	W410X53
19,18	6	W310X52	W460X82	W460X128	W410X53
16,15	9	W200X41.7	W200X22.5	W150X37.1	W200X22.5
14,13	15	W920X201	W760X161	W410X60	W310X28.3
12,11	21	W920X201	W920X345	W690X265	W310X32.7
10,9	29	W1000X258	W1100X499	W1000X412	W360X39
8,7	37	W250X73	W610X82	W360X39	W310X28.3
6,5	48	W840X193	W760X147	W760X196	W310X38.7
4,3	59	W1000X222	W1000X249	W760X257	W610X101
2,1	73	W1100X343	W1000X249	W1100X433	W610X101
Top Story Drift (cm)		8.954	8.76	9.576	10.02
Inter-Story Drift (cm)		0.75	0.744	0.733	0.748
Max. Strength Ratio		0.795	0.69	0.892	1
Weight (kN)		6259.736	6431.886	6337.728	4716.756

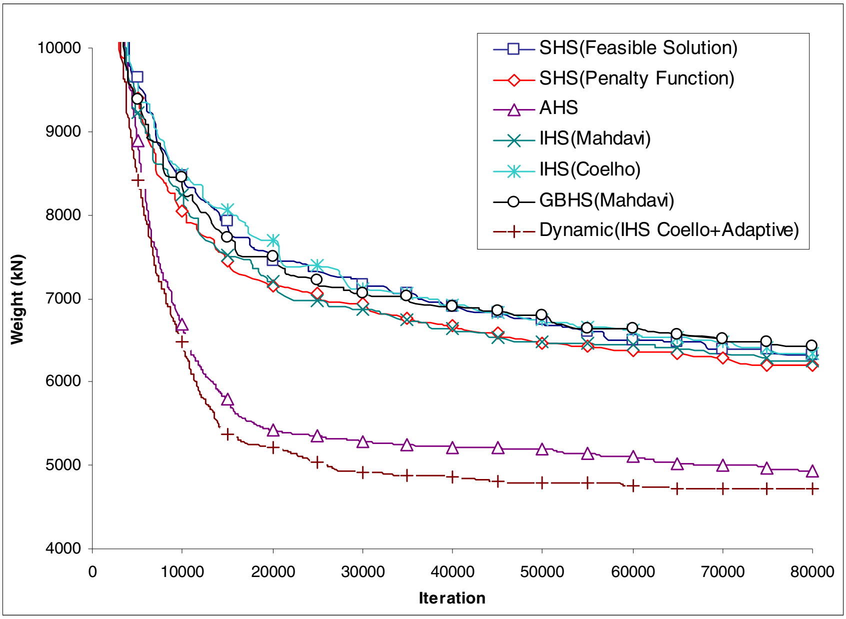


Fig. 6.11 Design histories of for twenty-story, 1860-member steel space frame

Table 6.7 Performance evaluation of seven different versions of the harmony search algorithms in the design examples

Design Examples	SHSAES	SHSPF	AHSPF	IHS (Mahdavi)	GBHS (Mahdavi)	IHSC (Coelho)	DHS Present Study
Five-story frame	5	3	**1**	4	7	6	2
Ten-story frame	4	6	2	3	7	5	**1**
Twenty-story frame	5	3	2	4	7	5	**1**

6.6 Conclusions

Seven different structural optimization algorithms are developed that are based on seven different versions of the harmony search algorithms that are recently developed. Three steel space frames are designed by these algorithms to evaluate their performance in finding the optimum solutions. All of these alternative harmony search algorithms are shown to be reliable, robust and effective algorithms. However, two versions among the all; adaptive harmony search method and dynamic harmony search method show better performance than the other versions. Particularly in the third design example where there are relatively large number of design variables and bigger design domain, the dynamic harmony search method has succeeded to find the optimum weight which is 25.36% less than the one determined by the standard harmony search algorithm. The performance evaluation of all these techniques in the design of three steel space frames considered is summarized in Table 6.7.

References

1. Load and Resistance Factor Design, Structural Members Specifications Codes, 3 edn.,vol.1 American Institute of Steel Construction (2001)
2. Lee, K.S., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. Computers and Structures 82, 781–798 (2004)
3. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering 194, 3902–3933 (2005)
4. Geem, Z.W., Lee, K.S., Tseng, C.-L.: Harmony Search for Structural Design. In: Proceedings of 2005 Genetic and Evolutionary Computation Conference (GECCO-2005), Washington, DC, USA, June 25–29 pp. 651–652 (2005)
5. Geem, Z.W. (ed.): Music-Inspired Harmony Search Algorithm; Theory and Applications, Studies in Computational Intelligence. SCI, vol. 191. Springer, Heidelberg (2009)
6. Geem, Z.W. (ed.): Harmony Search Algorithm for Structural Design Optimization. SCI, vol. 219. Springer, Heidelberg (2009)
7. Kochenberger, G.A., Glover, F.: Handbook of Meta-Heuristics. Kluwer Academic Publishers, Dordrecht (2003)
8. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys 35(30), 268–308 (2003)
9. Gonzales, T.F.: Handbook of Approximation Algorithms and Metaheuristics. CRC Press, Chapman & Hall (2007)
10. Yang, X.-S.: Nature Inspired Metaheuristic Algorithms. Luniver Press (2008)
11. Luke, S.: Essentials of Metaheuristic. Lulu Press (2011)
12. Erdal, F., Saka, M.P.: Optimum design of grillage systems using harmony search algorithm. In: Proceedings of 8th International Conference on Computational Structures Technology (CST 2006), Las Palmas de Gran Canaria, Spain September 12–15 (2006)
13. Saka, M.P.: Optimum design of steel sway frames to BS5950 using harmony search algorithm. In: Proceedings of The Eleventh International Conference on Civil, Structural and Environmental Engineering Computing (CC-2007) Civil-Comp Press, St. Julians (2007)
14. Saka, M.P.: Optimum geometry design of geodesic domes using harmony search algorithm. An International Journal, Advances in Structural Engineering 10(6), 595–606 (2007)
15. Carbas, S., Saka, M.P.: Optimum design of single layer network domes using harmony search method. Asian Journal of Civil Engineering, 10(1), 97–112 (2009)
16. Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P.: Performance evaluation of metaheuristic techniques in the optimum design of real size pin jointed structures. Computers and Structures 87, 284–302 (2009)
17. Saka, M.P.: Optimum design of steel sway frames to BS5950 using harmony search algorithm. Journal of Constructional Steel Research 65, 36–43 (2009)

18. Saka, M.P., Erdal, F.: Harmony search based algorithm for the optimum design of grillage systems to LRFD-AISC. Structural and Multidisciplinary Optimization 38, 25–41 (2009)
19. Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P.: Comparison of non-deterministic search techniques in the optimum design of real size steel frames. An International Journal, Computers and Structures 88(17-18), 1033–1048 (2010)
20. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 188, 1567–1579 (2007)
21. Omran, M.G.H., Mahdavi, M.: Global-Best harmony search. Applied Mathematics and Computation 198, 643–656 (2008)
22. Saka, M.P., Hasancebi, O.: Adaptive Harmony Search Algorithm for Design Code Optimization of Steel structures. In: Geem, Z.W. (ed.) Harmony Search Algorithms for Structural Design Optimization. SCI, vol. 239, pp. 79–120. Springer, Heidelberg (2009)
23. Hasançebi, O., Erdal, F., Saka, M.P.: An Adaptive Harmony Search Method for Structural Optimization. Journal of Structural Engineering, ASCE 136(4), 419–431 (2010)
24. Coelho, L.D.S., Bernett, D.L.D.A.: An improvement harmony search algorithm for synchronization of discrete-time chaotic systems. Chaos, Solutions and Fractals 41, 2526–2532 (2009)
25. Ad Hoc Committee on Serviceability: Structural Serviceability: A critical Appraisal and Research needs. Journal of Structural Engineering, ASCE 112(12), 2646–2664 (1986)
26. Ellingwood, B.: Serviceability guidelines for steel structures. Engineering Journal, AISC 26,1st Quarter 1–8 (1989)
27. Chen, W.F., Kim, S.-E.: LRFD Steel Design Using Advanced Analysis. CRC Press, Boca Raton (1997)
28. ASCE 7-05. Minimum Design Loads for Building and Other Structures (2005)
29. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: FromNatural to Artificial System. Oxford University Press, U.K (1999)
30. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan KaufmanPublishers, San Francisco (2001)