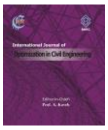
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(1):31-54



A COMPARATIVE STUDY FOR THE OPTIMAL DESIGN OF STEEL STRUCTURES USING CSS AND ACSS ALGORITHMS

A. Kaveh[1*, †], N. Khodadadi[1] and S. Talatahari[2]
[1]*School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran*
[2]*Department of Civil Engineering, University of Tabriz, Tabriz, Iran*

ABSTRACT

In this article, an Advanced Charged System Search (ACSS) algorithm is applied for the optimum design of steel structures. ACSS uses the idea of Opposition-based Learning and Levy flight to enhance the optimization abilities of the standard CSS. It also utilizes the information of the position of each charged particle in the subsequent search process to increase the convergence speed. The objective function is to find a minimum weight by choosing suitable sections subjected to strength and displacement requirements specified by the American Institute of Steel Construction (AISC) standard subject to the loads defined by Load Resistance Factor Design (LRFD). To show the performance of the ACSS, four steel structures with different number of elements are optimized. The results, efficiency, and accuracy of the ACSS algorithm are compared to other meta-heuristic algorithms. The results show the superiority of the ACSS compared to the other considered algorithms.

Keywords: Advanced charged system search; Optimal design; Steel structures; Opposition-based Learning; Levy flight.

Received: 10 November 2020; Accepted: 5 January 2021

1. INTRODUCTION

Finding optimal designs of steel structures has been an important issue for engineers at an affordable time taking into account both safety and economy. The construction of the structures using different types of sections meets the serviceability and strength requirements based on the code by the American Institute of Steel Construction (AISC).

[*]Corresponding author: School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran
[†]E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

Several scientists have studied the size optimization of these structures in the last two decades. Steel structures are among the most common structures in structural engineering. Therefore, the economical and safe design of these structures is very indispensable.

Many researchers have employed meta-heuristic algorithms for the optimal design of steel structures [1, 2]. To optimize the real size and rigidly linked steel strucures, Hasançebi et al. [3] employed several metaheuristic algorithms, such as Evolution Strategies (ES) based on the performance design. Camp et al. [4] adopted the ant colony (ACO) based on the observed capacity design. Kaveh and Abbasgholiha [5] have utilized the Big-Bang Crunch (BB-BC) method to optimize steel structures. For the first time, Perez and Behdinan [6] utilized the Particle Swarm Optimization (PSO) for the design of the steel frames. Carbas [7] improved the performance of the Firefly Algorithm (FA) and optimized two steel structures with this algorithm. Kaveh et al. [8] developed an efficient hybrid optimization algorithm based on Invasive Weed Optimization and Shuffled Frog-Leaping algorithm (SFLA-IWO) for optimal design of structures. Harmony Search (HS) was used by Saka to optimize the strucures [9]. The optimization of steel structures with an Enhanced Firefly Algorithm (EFA) is performed by Carbes [10]. Size optimization of real-size steel structures was investigated with Bat Inspired (BI) algorithms by Hasançebi and Carbas [11]. The HS algorithm is improved by Maheri and Narimani [12] and utilized to the optimal design of steel structures. The study of the different well-known algorithms to investigate the performance of them is focused by Alberdi and Khandelwal [13]. The optimization of steel braced structures considering the dynamic soil-structure interaction is performed by Bybordiani and Kazemzadeh Azad [14].

Charged system search (CSS) is a physical and mechanical optimization algorithm developed by Kaveh and Talatahari [15] for the design of structures. Kaveh and Zakian [16] developed a new version of the CSS called Adaptive Charged System Search (ACSS) algorithm and applied to solve economic dispatch cases. Kaveh et al. [17] used ACSS for real steel structures with new box-shaped sections for different examples. The performance of the ACSS algorithm is compared to that of the Upper Bound Strategy [18] integrated versions of the standard Big Bang-Big Church method and two newly developed editions; namely modified BB–BC (MBB–BC) [19] and exponential BB–BC (EBB–BC) [20] algorithms. This paper uses the idea of Opposition-based Learning (OBL) [21] and Levy Flight [22, 23] to enhance the optimization abilities and uses the information of each agent's new position in the subsequent search process to increase convergence speed of the ACSS.

The remaining paper is structured as: Section 2 presents an optimal steel structures design problem using the AISC-LRFD [24] regulations. The ACSS algorithm is presented in Section 3 after a brief introduction to the CSS. Section 4 comprises four different structures and their optimum results. Finally, the paper ends in Section 5 with conclusions.

2. OPTIMAL DESIGN OF STEEL STRUCTURES CONSIDERING AISC-LRFD CODE

In practical applications, structures members are chosen from a set of existing steel sections which yields a discrete sizing optimization problem. In this research, the optimal design of structures is formulated as

$$\begin{array}{ll} find & X = [x_1, x_2, \ldots, x_n] \\ to\ Minimize & fit(X) = W(X) \times f_{penalty}(X) \end{array} \quad (1)$$

where, *X* is the vector of design variables containing the cross-sectional areas of *W* sections; the number of design variables is described by *n*; the fitness function is considered as *fit*(*X*); *W*(*X*) is the cost function of the structure; $f_{penalty}(X)$ is the penalty function which results from the violations of the constraints corresponding to the response of the structure. The cost function in the form of the weight of the structures is formulated as

$$W(X) = \sum_{k=1}^{n} \mathrm{\rho_i \cdot A_i \cdot L_i} \quad (2)$$

where, the unit weight of the steel section is defined as $\mathrm{\rho_i}$; the cross and the length of the steel section are described as A_i and L_i, respectively.

Here, several development restrictions, including strength and serviceability criteria, are imposed under the objective of evaluating the minimum structural costs. The following design constraints (C_{IEL}^{i} and C_{IEL}^{v}) should be satisfied with strength necessity, according to the AISC-LRFD [24] practice code.

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

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

$$C_{IEL}^{i} = (V_{uj})_{IEL} - (\phi_v V_n) \leq 0 \quad (5)$$

In Eqs. (3- 5), *IEL* is the element number; *NEL* is the total number of elements; *J* is the number for load combination and *N* is the entire number of load design combinations. P_{uj} is the required axial (tensile or compressive) strength, under the *j*th design load combination. M_{uxj} and M_{uyxj} are the required flexural strengths for bending about *x* and *y* under the *j*th layout load combination, respectively; where subscripts *x* and *y* are the relating symbols for strong and weak axes bending, respectively. Then again, 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 evaluation. ϕ is the resistance factor for axial strength, which is 0.9 for both compression and tension (based on yielding in the gross section) and ϕ_b is the resistance factor for flexure, which is equal to 0.9. Here, Eq. (4) 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 IELth member under evaluation. The nominal shear strength is multiplied by a resistance variable in an attempt to calculate the construction shear strength ϕ_b of 0.9 [24].

The serviceability constraints in the design process should be evaluated in addition to the strength criteria. In this study, serviceability requirements (C_D^t and C_F^d) are proposed as follows

$$C_D^t = \Delta_{MaxJ} - \Delta_{Max}^a \leq 0 \tag{6}$$

$$C_F^d = \left[\delta_J\right]_F - [\delta^a]_F \leq 0 \tag{7}$$

Eq. (6) compare the lateral structure maximum displacement in the Dth direction under the jth load combination, i.e. Δ_{MaxJ}, with maximum permissible lateral displacements, i.e. Δ_{Max}^a. Likewise, Eq. (7) controls the inter-story drift of the Fth story under the jth load combination, $[\delta_J]_s$, against the associated authorized value, $[\delta^a]_s$. Here, NF is the number of stories.

3. THE UTILIZED OPTIMIZATION ALGORITHM

3.1 Standard charged system search

The charged search system is based on the electrostatic and Newtonian mechanics laws [26]. Inside and outside the electric field (E_{ij}) of the charged insulating solid sphere, the Coulomb and Gauss law provide the magnitude of the electric field as follows:

$$E_{ij} = \begin{cases} \dfrac{K_e q_i}{a^3} r_{ij} & if \quad r_{ij} < a \\ \dfrac{K_e q_i}{r_{ij}^2} & if \quad r_{ij} > a \end{cases} \tag{30}$$

where, the constant known as the Coulomb constant is defined by K_e; the sphere center separation and the chosen point is described by r_{ij}; the magnitude of the charge is described as q_i and the radius of the charged sphere is defined by a. Using the superimposition principle, the resulting electric force due to N charged spheres (F_j) is equal to:

$$F_j = k_e q_j \sum_{i,j\neq i}^{N} \left(\frac{q_i}{a^3} r_{ij} . i_1 + \frac{q_i}{r_{ij}^2} . i_2\right) \frac{r_i - r_j}{\|r_i - r_j\|} \qquad \begin{cases} i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{cases} \tag{31}$$

Also, according to to the mechanics of Newtonian, we have:

$$\varDelta r = r_{new} - r_{old} \tag{32}$$

$$v = \frac{r_{new} - r_{old}}{t_{new} - t_{old}} = \frac{r_{new} - r_{old}}{\varDelta t} \tag{33}$$

$$a = \frac{v_{new} - v_{old}}{\varDelta t} \tag{34}$$

where, r_{old} and r_{new} are the initial and final positions of the particle, respectively; The particle velocity is described as v and a is the particle acceleration. By using Newton's second law, the displacement of any object in the function of time is achieved by combining the above equations:

$$r_{new} = \frac{1}{2}\frac{F}{m}.\varDelta t^2 + v_{old}.\varDelta t + r_{old} \tag{35}$$

Fig. 1. depicts the flowchart of the CSS algorithm based on electrostatic and Newtonian mechanical laws [26].

3.2 Advanced charged system search

The Advanced Charged System Search (ACSS) algorithm is consided to improve the performance of the CSS in the current section. Four effective steps for the design of the ACSS are considered in this study. At the first, the initialization is based on Opposite-based Learning and sub-spacing techniques. The next one is to use previous information directly after their generation. The third one is on applying the Levy flight random walk to enrich the algorithm updating process. The last one is about applying the position information of each agent for a subsequently charged particle for improving the speed of convergence. The power of rules proliferates and the power of randomization is diminished by improving the optimization process.

3.2.1 Basic Stages of the Advanced Charged Search

The stages of the ACSS are described as follows:

Stage 1. Initialization:

CP arrays or initial positions are randomly determined in the search area, while their initial speed is set to zero. The fitness magnitudes for the CPs are calculated and theey are classified ascending. The best CP in the entire group of CPs is considered as *Xbest* with the best fitness (*fitbest*). Similarly, the worst one is considered as *fitworst*.

Two main features of metaheuristics are problem independence and low reliance on initialization. However, defing better intail results will help the algorithms to improve their performance. On the other hand, by adding three new spaces to the CSS, which uses an initializing search space, the ACSS uses a 4-fold initialization space. In the first step, the ACSS performs similar to the CSS. Next, the idea of Opposition-based Learning (OBL) is utilized to initialize space number two. OBL, the idea of soft computing to speed up convergences between diverse optimizers, has been designed by Tizhoosh et al. [21]. The OBL employs the population and its opposite counterparts to develop better potential solutions. Many studies have shown that OBL offers the opportunity to find the global best [27]. Optimal solutions tend to be close domain boundaries thus, the upper and lower bonded domains can be separated to satisfy these conditions. Two remain spaces are initialized from the lower and upper limited subdomains. Therefore, there are four spaces in the initialization of ACSS, as:

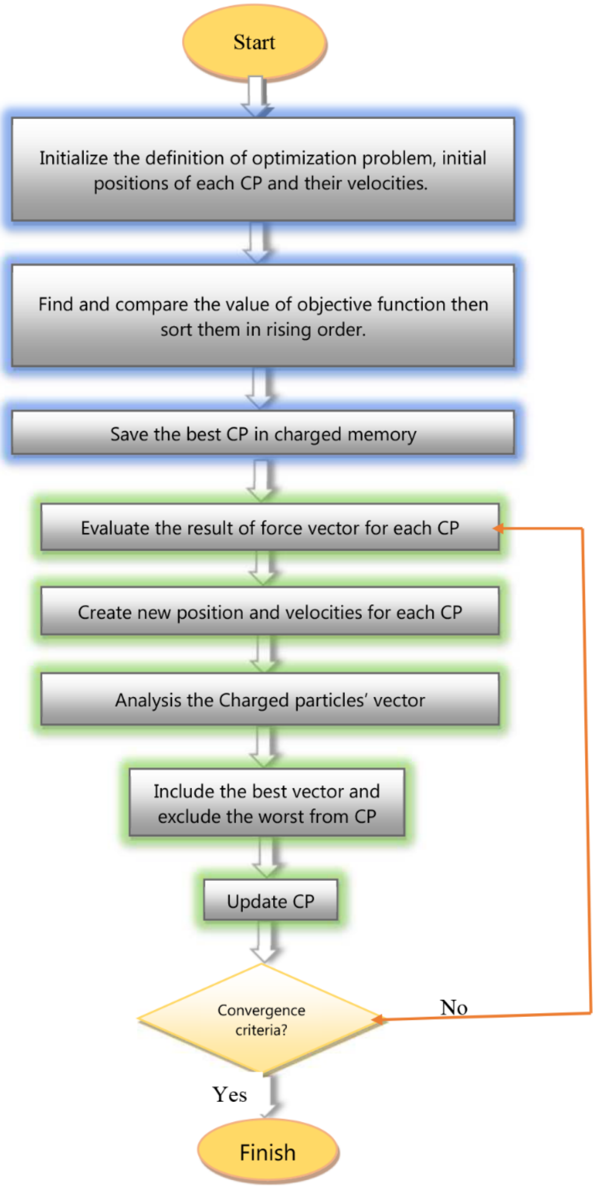

Start
Initialize the definition of optimization problem, initial positions of each CP and their velocities.
Find and compare the value of objective function then sort them in rising order.
Save the best CP in charged memory
Evaluate the result of force vector for each CP
Create new position and velocities for each CP
Analysis the Charged particles' vector
Include the best vector and exclude the worst from CP
Update CP
Convergence criteria?
No
Yes
Finish

Figure 1. Flowchart of the CSS

First space:

$$x_{i,j}^{Initial1} = x_{i,\min} + rand.(x_{i,\max} - x_{i,\min}) \qquad i = 1,2,3,\ldots,nv \tag{36}$$

Second space:

$$x_{i,j}^{Initial2} = x_{i,\max} + x_{i,\min} - x\,_{i,j}^{Initial1} \qquad i = 1,2,3,\ldots,nv \tag{37}$$

Third space:

$$x_{i,j}^{Initial3} = x_{i,\min} + rand.\left(\frac{x_{i,\max} + x_{i,\min}}{2} - x_{i,\min}\right) \quad i = 1,2,3,\ldots,nv \tag{38}$$

Fourth space:

$$x_{i,j}^{Initial4} = \frac{x_{i,\max} + x_{i,\min}}{2} + rand.\left(x_{i,\max} - \frac{x_{i,\max} + x_{i,\min}}{2}\right) \quad i = 1,2,3,\ldots,nv \tag{39}$$

In the Charged Memory (CM), the best solutions from the four spaces are now saved without changing the CM size as defined in the standard CSS. CM is a matrix that keeps several of the best CPs and their fitness values. *Rand* is the random number distributed uniformly from [0,1].

Stage 2. Solution construction:

❖ *Forces Determination*. The force vector is evaluated for the jth CP as following:

$$F_j = q_j \sum_{i,j\neq i} \left(\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{r_{ij}^2}.i_2\right) P_{ij}(X_i - X_j) \qquad \begin{cases} j = 1, 2, \ldots, N \\ i_1 = 1, i_2 = 0 \;\Leftrightarrow\; r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow\; r_{ij} \;\geq a \end{cases} \tag{40}$$

❖ Creation New Solution. Each CP goes to the new position as following:

$$X_{j,new} = rand_{j1}.k_a.\frac{F_j}{m_j}.\Delta t^2 + rand_{j2}.k_v.v_{j.old}.\Delta t + X_{j,old} \tag{41}$$

$$v_{j,new} = \frac{X_{j,new} - X_{j,old}}{\Delta t} \tag{42}$$

After moving the CP to its new position, the objective function is determined. In the current algorithm, iteration changes continuously and all updating processes occur after generating only one solution. This means that the current location of each agent using the ACSS algorithm will affect the movement of subsequent CPs. However, the new positions in the standard CSS can not be utilized unless the current iteration is completed [28].

Stage 3. Implemented Levy flight

It is proposed that the Levy flight algorithm enhances a random exploration. The Levy flight is an efficient random route recently implemented efficiently in optimization methods [22, 23]. The Levy movement is a non-Gaussian discrete method based on Levy as a power-law formulation:

$$L(s) \approx |s|^{-1-\beta} \tag{43}$$

where, β is defined as the stability parameter in the range (0, 2). A simple version of Levy distribution can be ruled out in the mathematical example:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} exp[-\frac{\gamma}{2(s-\mu)}] \frac{1}{(s-\mu)^{\frac{3}{2}}} & if \quad 0 < \mu < s < \infty \\ 0 & if \quad s \leq 0 \end{cases} \tag{44}$$

in which, μ is the shift parameter $\gamma > 0$; parameter is the distribution control scale and the skewness parameter is prescribed by α within the interval [-1,1].

In this context, Levy flight is incorporating the influence of the best local solution, X_{best}, to improve the algorithm. By incorporating Levy Flight to the update procedure, the ACSS will have a new location as following [16]:

$$X_{j,new} = rand_{j1}.F_j + rand_{j2}.k_v.V_{j,old} + rand_{j3}.\alpha.Levy(\beta) + X_{j,old} \tag{45}$$

while the new velocity is defined as:

$$V_{j,new} = X_{j,new} - X_{j,old} \tag{46}$$

The step size is the scale of the problem and the non-trivial step forming scheme is described as:

$$rand_{j3}.\alpha.Levy(\beta) \approx 0.01 \frac{u}{|\nu|^{\frac{1}{\beta}}} (X_{best} - X_{j,old}) \tag{47}$$

when u and μ are chosen using normal distribution, as:

$$u \approx N(0,\sigma_u^2), \;\; \nu \approx N(0,\sigma_v^2) \tag{48}$$

in which, $rand_{j1}$, $rand_{j2}$ and $rand_{j3}$ are uniformly distributed random numbers in the range [0,1]. This approach implements the impacts of the best CPs during the update procedure. Furthermore, the ACSS has no acceleration coefficient k_a, and it only has k_v (like standard CSS). k_v is defined as a decreasing function for stabilizing the effects of the previous speed and controlling the exploration procedure, as:

$$k_v = c_v \times [1 - \left(\frac{iter}{iter_{max}}\right)] \tag{46}$$

where, *iter* is the current iteration number and $iter_{max}$ is the maximum iteration number. *cv* has steady values to adapt to the problem of optimization [29]. When the new CP exits the feasaible search space during the updating process, similar to the CSS, a harmony search-based handling method may be employed to rectify its position oriented on the search space of the issue conducted. This strategy requires the regeneration of any solution vector component violating the variable limits from a CM or a random value form the possible range. Besides, if some new CP-vectors are better than those in CM, the worst ones in CM are replaced and the worst vectors are removed.

Stage 4. Terminating criterion control:

❖ Stages 2 and 3 are repeated until a final criterion has been fulfilled.

4. NUMERICAL EXAMPLES

In this section, the efficiency of the new algorithm is investigated. To fulfill this aim, the optimization process is carried out using the CSS and ACSS algorithms and tested with four optimization examples of steel structures [18]. Each problem is solved 30 times independently. Both CSS and ACSS are employed the same number of analyses and agents to compete reasonably.

The algorithm has been coded in MATLAB 2020b and the structures are analyzed with a direct stiffness method by SAP2000 v14.1 using an application programming interface (API) throughout the optimization process. The current work is done with the following features on the computer: CPU 2.3 GHZ (an Intel Core i9 computer platform), Ram 16 GB 2400 MHz DDR4 with Macintosh (macOS Big Sur).

There are four examples in this paper: a 942-bar spatial truss, a 12 *m*×12 *m* grillage system, a 3-story steel frame with 147 members and a 10-story with 1026-members. Results of each problem are then compared to those obtained by other methods.

4.1 A 942-bar spatial truss

A 26-story-tower space truss containing 942elements and 244 nodes is considered as the first example. Fifty-nine design variables are used to represent the cross-sectional areas of element groups in this structure, employing the symmetry of the structure. Fig. 2 shows the geometry and the 59 element groups. This example has been optimized using 6 meta-heuristic algorithms, previously. The CSS+PSO and CSS method achieved good solutions after 15,000 nalyses and found optimum weights of $46{,}310\,(205{,}997\,N)$ and $47{,}371\ lb\ (210{,}716\ N)$, respectively [30]. The best weights for the GA, PSO, BB–BC and HBB–BC were 56,343 *lb* (250.626*N*), 60.385*lb* (268.606*N*), 201.00*lb* (236.650*N*) and 52.401*lb* (233.091*N*), repectively [31]. The new algorithm can find the best result among others as shown in Table 1. The best result of this new algorithm is equal to

$45{,}965\ lb\ (2044{,}62\ N)$. The new algorithm has better performance in terms of the optimization time, standard deviation and the average weight. It converges to a solution after 10,000 analyses of structures in average.

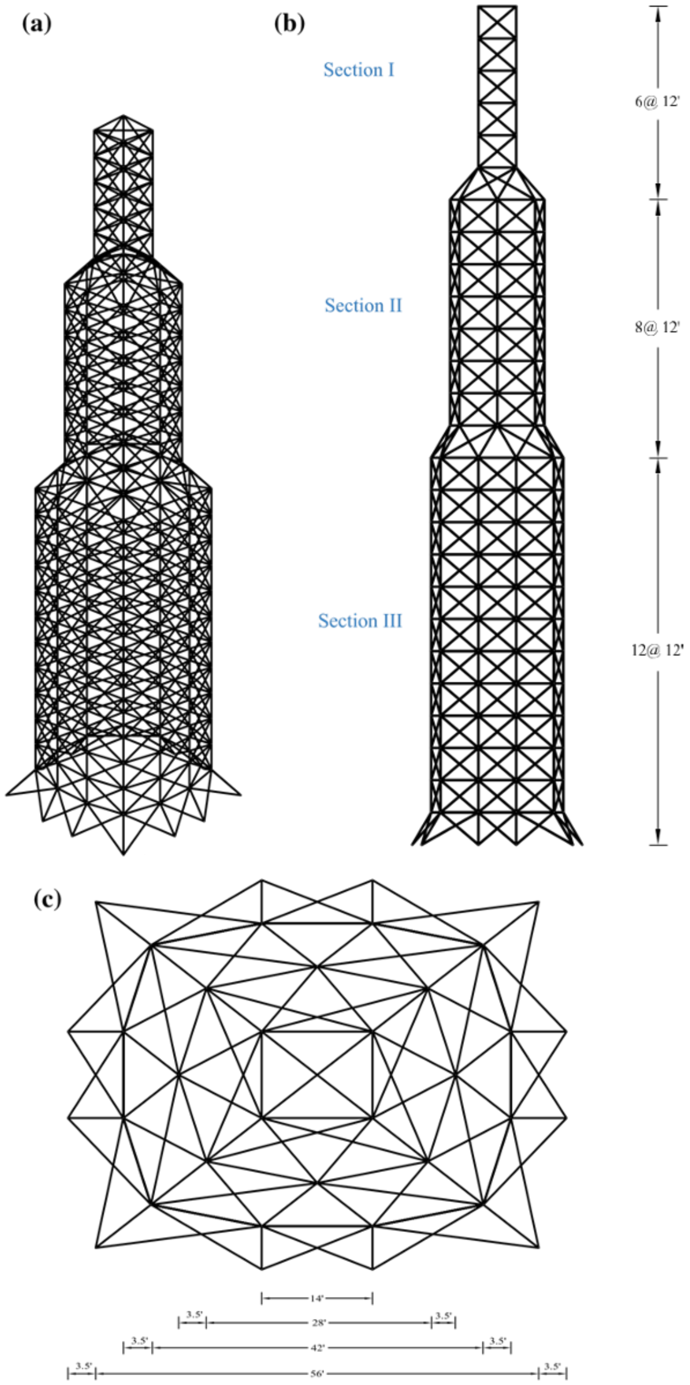


Figure 2. A 942-bar spatial truss

Table 1: Performance comparison for the truss example

Methods	**GA [31]**	**PSO [31]**	**BB–BC [31]**	**HBB–BC [31]**	**CSS [32]**	**CSS+PSO [32]**	**ACSS**
Best weight (lb)	56343	60385	53201	52401	47371	46310	45965
Average weight (lb)	63223	75242	55206	53532	48603	47953	47530
Std Dev (lb)	6640.6	9906.6	2621.3	1420.5	950.4	874.3	725.5
No. of analyses	50,000	50,000	50,000	30,000	15,000	13,500	10,000
Optimization time (sec.)	4,450	3,640	3,162	1,926	1,340	1,190	925

4.2 A 12m×12m grillage system

As the second example, a grillage system with a $12\,m \times 12\,m$ square area is considered. The system is supposed to carry a $15\,kN/m^2$ uniformly distributed load (total load is $2{,}160\,kN$). The grillage system that can be used to cover the area has the longitudinal beams of length $12\,m$ and the transverse beams of length $12\,m$. This system is composed of $2\,m$ beams as shown in Fig. 3, where the system has 60 members. The total external load is distributed on the joints of the grillage system as point loads.

The vertical displacements of middle joints are restricted to $25\,mm$. When four group designs are considered in such a way that the outer and inner longitudinal beams belong to the group 1 and 2, respectively, while the outer and inner transverse beams are taken as the group 3 and 4, respectively. The weight obtained by the new algorithm is $9{,}211\,kg$ same as the reported one by CSS+PSO [30] while it has been $9{,}251\,kg$ for the CSS method [32]. The optimum results obtained by the new algorithm as well as the hybrid CSS and the standard CSS are summarized in Table 2. The number of required structural analyses for this example was equal to 1.500 which is less than 2.560 and 3.000 analyses required for the CSS+PSO and standard CSS, respectively.

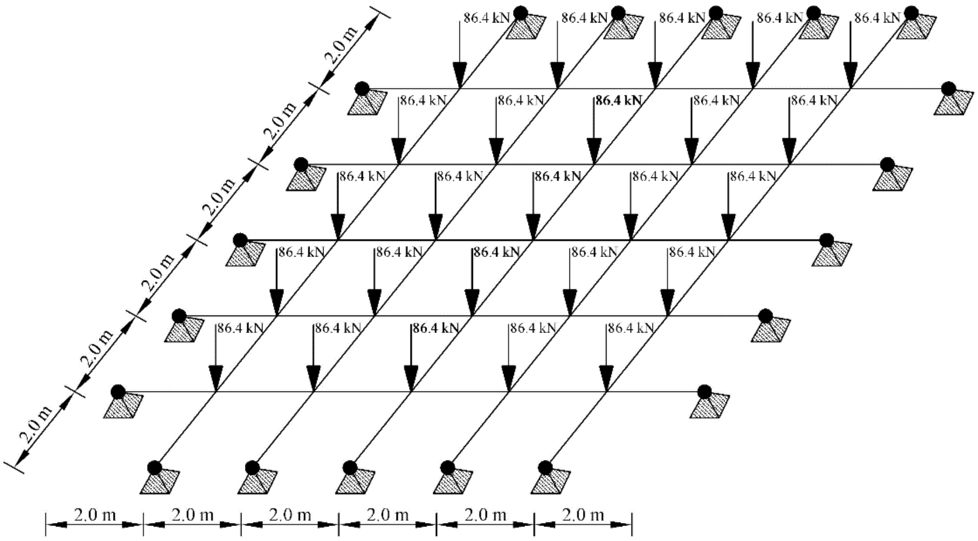


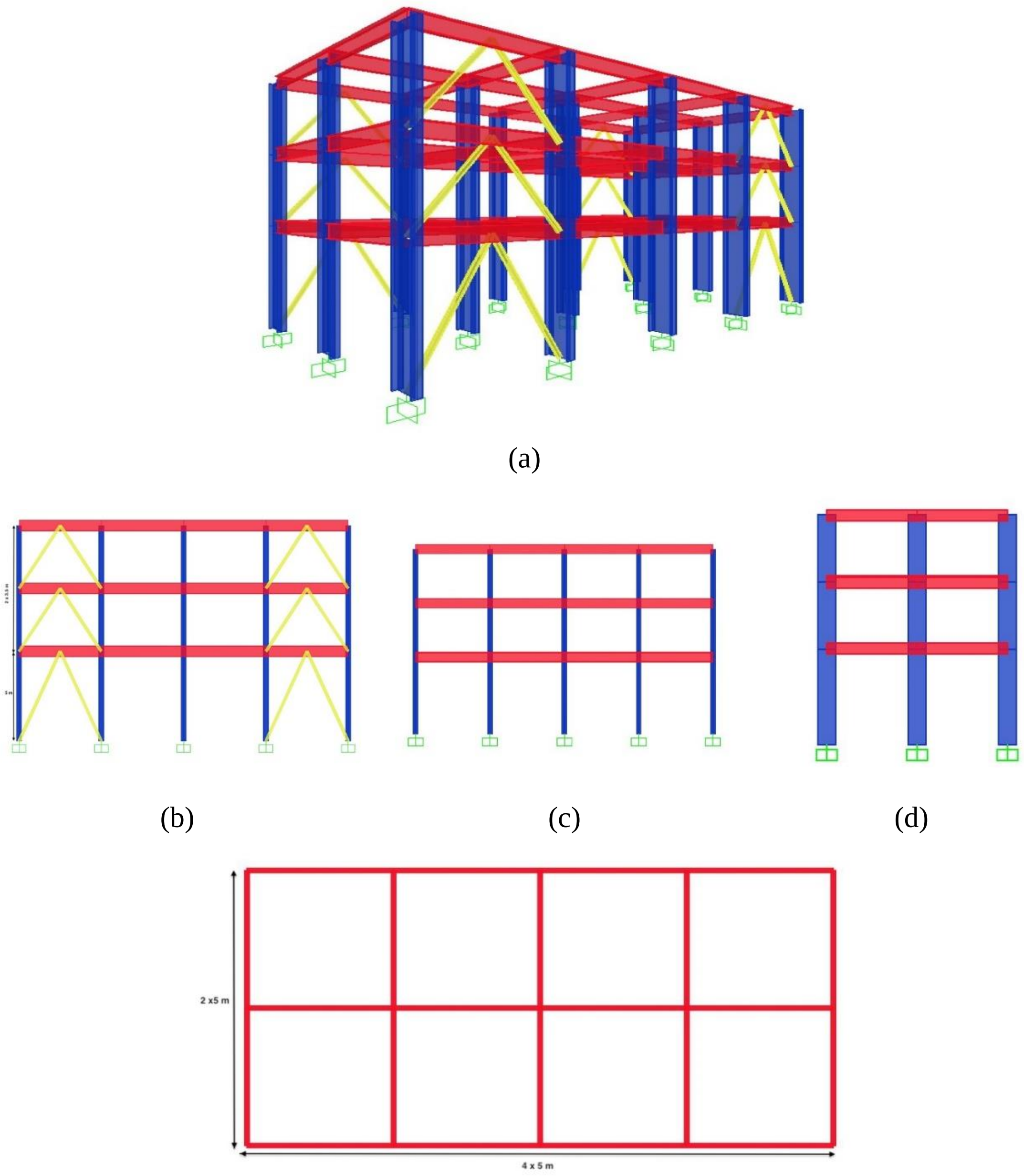
Figure 3. A 60-elements grillage system

Table 2: Optimal design for the grillage system example

Element Group	Optimal W-shaped sections		
	CSS [32]	**CSS+PSO [30]**	**ACSS**
1	W6X9	W10X12	W10X12
2	W36X135	W36X135	W36X135
3	W12X14	W8X10	W8X10
4	W12X22	W14X22	W14X22
Weight (kg)	9,251	9,211	9,211
δ^u (mm)	24.3	23.4	23.4
Maximum Strength Ratio	99.0%	99.2%	99.2%

4.3 A 3- story steel frame

The frame with three stories depicted in Fig. 4 is considered as the third example. This structure has 147 members, including 78 beams, 24 bracing elements, and 45 columns. The stability of the structure is ensured by the use of moment-resistant connections in addition to inverted V-type bracing systems next to the x-direction. The 147 frame members are collected in 10 member groups for requirements of practical manufacturing, as shown in Fig. 5. In the plane level, four sizing variables are grouped as corner columns, interior, side x-z, and side y-z, and it is assumed that the three stories are contained from the same cross-section. From a different point of view, each story has three design variables for the frame which group all beams into one sizing variable. Likewise, each story is composed of a single variable of size (bracing) which leads to three variables for the frame design. It should be noted that in the stage of analysis, the floor plates are not modeled [17]. The 10 load combinations are considered as design loads as presented in Table 3; in which, the dead and live loads are defined as, D and L, respectively; earthquake loads E_x and E_y, applied to the center of mass in x and y, respectively. The earthquake loads E_{ex} and E_{ey} are applied to the center of mass in x and y directions, taking into consideration the effect of accidental eccentricity, respectively. Based on AISC [24], depending on the direction of the earthquake load being applied, the eccentricity amount is 5% of the structural dimension. There are live loads of 12 and 7 kN/m on the floor and the roof beams, respectively. In addition, the distributed uniformly dead load on the floor and roof are considered as line loads on beams with an amount of of 20 and 15 kN/m, respectively. The structural weight is also taken into account. Earthquake loads are calculated based on AISC [24] lateral force equivalent. Here, the earthquake base shear (V_b) is taken as where $V_b = 0{,}15\, W_s$ and the entire dead load of the structure is W_s.



(a)

(b) (c) (d)

(d)

Figure 4. The 3-story with 147-member steel frame, (a) 3-D view (b) side view of frames 1, 3 (c) side view of frame 2 (d) side view of frames A, B, C, D and E (e) plan view

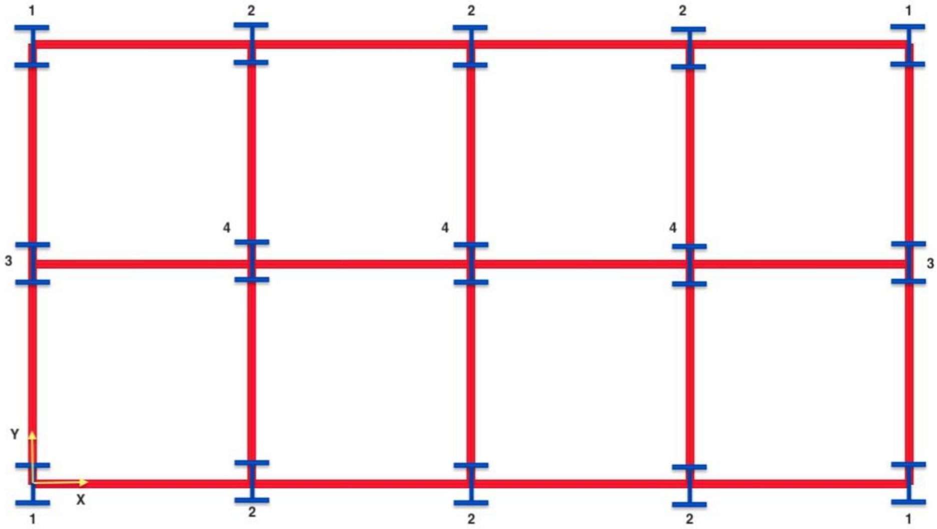


Figure 5. Columns grouping of the 3-story with 147-member steel frame in plan level

Table 3: Load Combinations

No.	Load Combination	No.	Load Combination
1	$1,4D$	6	$1,2D + 1,0E_y + 0,5L$
2	$1,2D + 1,0E_{ex} + 0,5L$	7	$0,9D + 1,0E_{ex}$
3	$0,9D + 1,0E_x$	8	$1,2D + 1,0E_x + 0,5L$
4	$0,9D + 1,0E_{ey}$	9	$1,2D + 1,0E_{ey} + 0,5L$
5	$1,2D + 1,6L$	10	$1,9D + 1,0E_y$

The lateral maximum displacement of the top level is restricted to $0{,}03m$, and the top of the story drift is considered as $h_t/400$, where h_t is the story's height. The drifts between the stories are evaluated by the displacement of every story's center. In terms of the maximum lateral movement at the top level, the maximum displacement of the structure ends is calculated. Here, all joint displacements of each story are constrained horizontally based on a rigid assumption of the diaphragm. In the current study, the optimum weight of the steel structure is carried out. Results show that the ACSS obtains the optimal design with a weight of $37{,}212\ ton$. Fig. 6. indicates the convergence history plots for both CSS and ACSS algorithms. For comparison, the ACSS found the optimal design with an average weight of $39{,}167\ ton$ and a standard deviation of 1,214 and the CSS algorithm finds the optimal design with an average weight of $40{,}964\ ton$ with a standard deviation of 1,571 (see Table 4). The ratio displacement of the roof is obtained 0,873 and 0,886 for the CSS and ACSS, respectively. The maximum stress ratio is 0,9533 and the inter-story drift to the allowable 0,9468 for ACSS according to Figs. 7 and 8.

Table 4: Comparison of the optimized 3- story steel frame with 147-members obtained by CSS

and ACSS with other algorithms

Group	**UBB-BC [25]**	**UMBB-BC [25]**	**UEBB-BC [25]**	**Present Work**	
				CSS	**ACSS**
CG_1^a	W 10x39	W 30x90	W 21x62	W 24x55	W 14x34
CG_2	W 27x84	W 14x48	W 14x48	W 24x68	W 21x62
CG_3	W 40x149	W 40x215	W 36x150	W 14x48	W 24x68
CG_4	W 18x65	W 27x84	W 21x68	W 24x68	W 21x62
B_1^a	W 21x44	W 14x34	W 18x40	W 18x35	W 21x50
B_2	W 16x40	W 12x35	W 18x35	W 21x44	W 18x40
B_3	W 10x22	W 18x35	W 16x26	W 16x31	W 12x26
BR_1^a	W 27x84	W 21x44	W 8x24	W 8x24	W 8x24
BR_2	W 16x26	W 10x22	W 16x26	W 6x15	W 6x15
BR_3	W 21x44	W 6x15	W 6x15	W 6x15	W 8x18
Weight (ton)	47.3	45.67	38.91	37.849	37.212

[a] CG denotes Column Group concerning Fig. 5, B: Beams, BR: Bracing

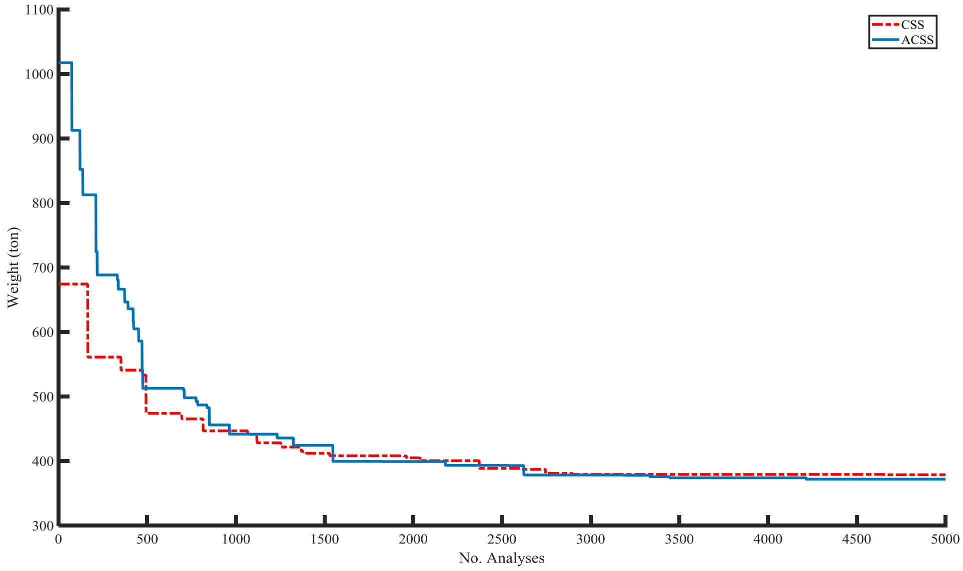


Figure 6. Convergence histories of the 3-story with 147-Member steel with ACSS and CSS algorithm

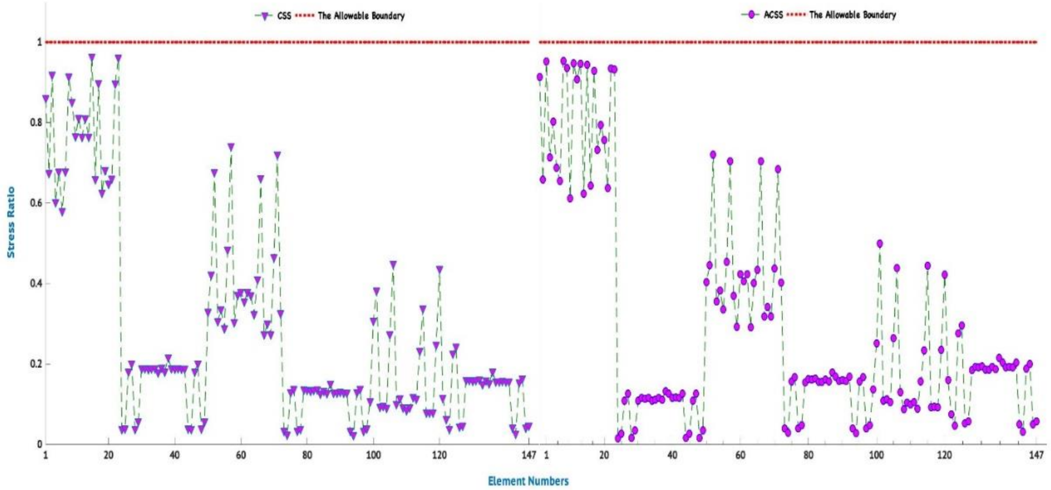


Figure 7. The elements stress ratio in the optimum frame design of the 3-story with 147-Member steel with ACSS and CSS algorithm

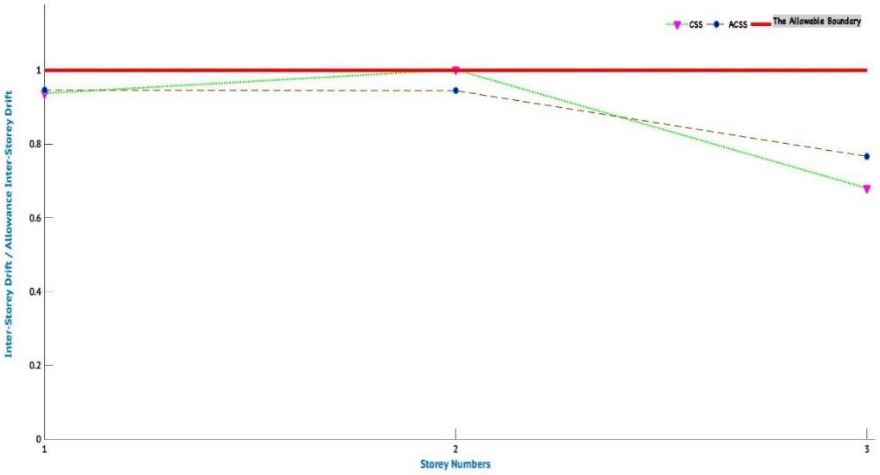
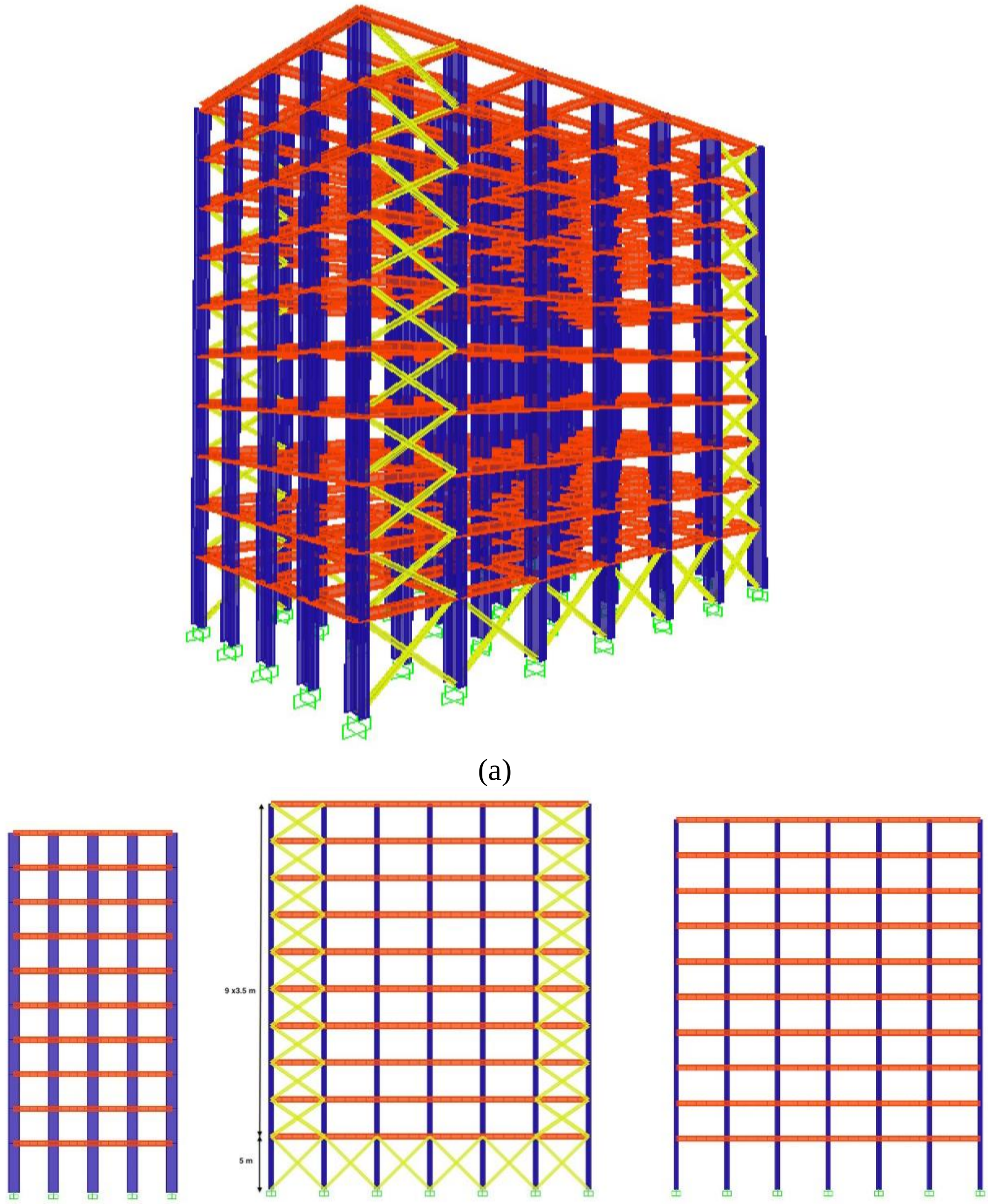


Figure 8. The ratio of the inter-story drift to the allowable inter-story drift in the optimum frame design for the 3-story with 147-Member steel using the ACSS and CSS algorithms

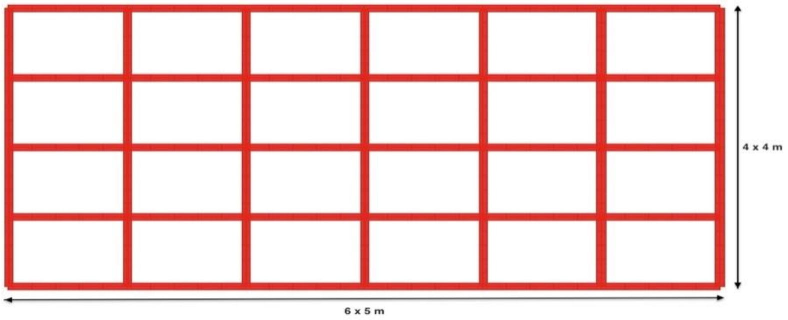
4.4 A 10-story frame

The frame with ten stories depicted in Fig. 9 is considered as the last example. This structure has 1026 members, including 580 beams, 96 bracing elements, and 350 columns. Besides inverted X-type bracing systems next to the *x*-direction, the stability of the structure is ensured by the use of moment-resistant connections. The 1026 frame members are collected into 32 member groups for requirements of practical manufacturing. The group of members is conducted in plan and level. The structural members in elevation are grouped into three stories except the first. On the plan level, columns in five groups as shown in Fig. 10 are defined; The beams are classified as outer and inner beams in two groups; bracings in one group are supposed to be. Thus, there is a total of 20 column groups, 8 beam groups, and 4 bracing groups based on groupings of elevation and plan level. It should be noted that in the stage of analysis the floor plates are not modeled [18].



(a)

(b) (c) (d)



(e)

Figure 9. The 10-story with 1026-Member steel frame, (a) 3-D view (b) side view of frames 2, 3 and 4 (c) side view of frames 1 and 5 (d) side view of frames A, B, C, D, E, F and G (e) plan view

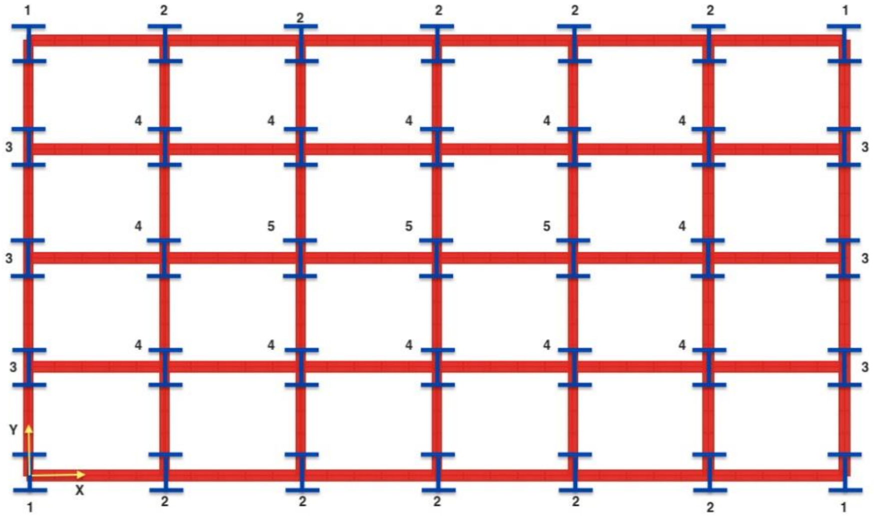


Figure 10. Columns grouping of the 10-story with 1026-Member steel frame in plan level

The ten load combinations are presented design as shown in Table 3. There are live loads of 12 and $7\ kN/m$, respectively on the floor and the roof beams, respectively. In addition to loads that are distributed uniformly on floor and roof beams with a load of 20 and $15\ kN/m$ respectively, the structural weight is also taken into account. The earthquake loads are calculated according to the same procedure as the first example. Here, the earthquake base shear (V_b) is taken as $V_b\ =\ 0{,}1\ W_s$.

The optimal structural weight was accomplished using the UBB–BC [18], UMBB–BC [18] and UEBB–BC [18] algorithms before and in the current study, the CSS and ACSS algorithms are utilized. Convergence curves are indicated in Fig. 11 for these algorithms. As reported by [17], 25.000 analyses are needed for UBB–BC, UMBB–BC and UEBB–BC algorithms to converge, while 21.000 analyses are sufficient for the CSS and ACSS algorithms. Table 5 lists the 10-story frame design for both CSS and ACSS and other optimization methods. It is important to note that the best ACSS design is a frame that weighs $540,384\,ton$, which is $21,32\%, 18,51\%, 4,36\%$ lighter than UBB–BC [18], UMBB–BC [18], UEBB–BC [18] design, respectively; Therefore, the ACSS algorithm gained the best result. The stress ratio of all members and the inter-story drift are calculated for the best design and are shown in the Figs. 12 and 13. The ratio displacement of the roof is $0,812$ for the CSS and $0,878$ for the ACSS. In this example, the average weight of the ACSS designs is $581,746\ ton$ with a standard deviation equal to $21,75$, while these numbers are $601,212\ ton$ and $38,17\ ton$ for the CSS. It is clear that that the tandard deviation and average values for the ACSS are lower than those of the CSS, indicating lower scattering of the ACSS solution and it can also be shown that the ACSS will find the best design.

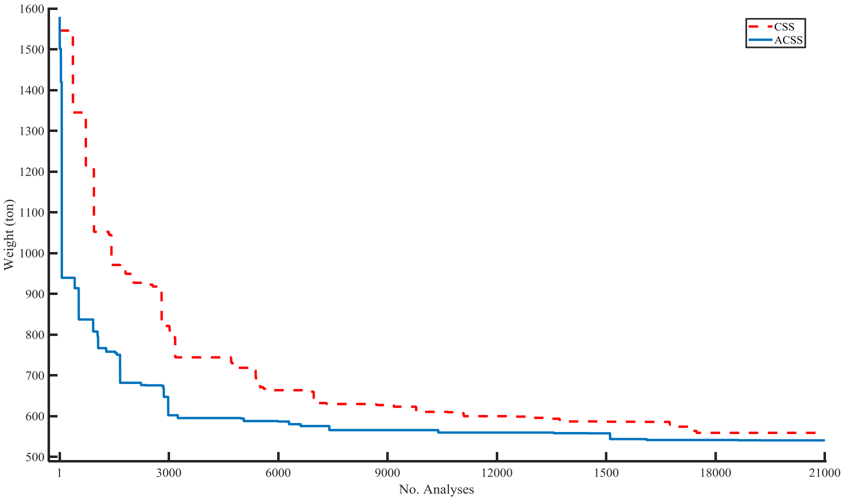


Figure 11. Convergence histories of the 10-story with 1026-Member steel with ACSS and CSS algorithm

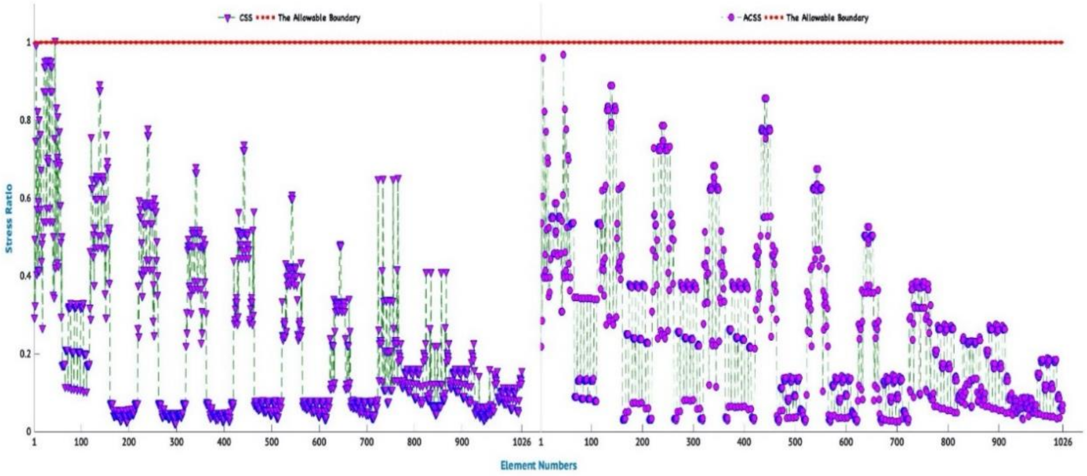


Figure 12. The elements stress ratio in the optimum frame design of the 10-story with 1026-Member steel with ACSS and CSS algorithm

Table 5: Comparison of the optimized 10- story steel frame with 1026-members obtained by the CSS and ACSS with other algorithms

Stories	**Groups**	**UBB-BC [25]**	**UMBB-BC [25]**	**UEBB-BC [25]**	**Present Work**	
					CSS	**ACSS**
1	CG_1^a	W 27x258	W 24x492	W 33x201	W 27x368	W 40x431
	CG_2	W 27x161	W 27x146	W 24x146	W 40x183	W 44x230
	CG_3	W 27x102	W 21x101	W 24x104	W 27x146	W 40x211
	CG_4	W 27x146	W 27x161	W 40x174	W 40x149	W 27x194
	CG_5	W 27x146	W 27x258	W 40x321	W 12x152	W 36x230
	IB^a	W 27x84	W 21x44	W 27x84	W10X33	W 21x44
	OB^a	W 27x84	W 27x84	W 27x84	W 16x40	W 8x24
	BR^a	W 27x94	W 30x90	W 18x76	W12x30	W 8x24
2-4	CG_1^a	W 27258	W 21x201	W 36x328	W 40x297	W 12x 230
	CG_2	W 27x146	W 24x162	W 36x245	W 30x148	W 30x235
	CG_3	W 27x84	W 24x131	W 36x135	W 40x149	W 33x241
	CG_4	W 27x102	W 40x174	W 33x118	W 24x146	W 30x 116
	CG_5	W 27x114	W 27x102	W 44x262	W 10x100	W 21x132
	IB^a	W 27x84	W 27x84	W 16x26	W 27x102	W 14x22
	OB^a	W 27x84	W 30x90	W 36x135	W 24x68	W 33x118
	BR^a	W 27x84	W 40x149	W 21x62	W 10x60	W10x49
5-7	CG_1^a	W 27x161	W 40x235	W 27x258	W 27x129	W 33x263
	CG_2	W 27x114	W 24x131	W 18x106	W 14x159	W 18x143
	CG_3	W 27x84	W 30x90	W 33x130	W 30x108	W 27x94
	CG_4	W 27x84	W 18x86	W 27x94	W 14x120	W 10x77
	CG_5	W 27x99	W 14x90	W 24x192	W 21x93	W 24x103
	IB^a	W 27x84	W 21x44	W 21x44	W 21x73	W 21x44
	OB^a	W 27x84	W 30x108	W 21x73	W 24x68	W 33x130
	BR^a	W 27x94	W 33x118	W 30x90	W 10x49	W12x53
8-10	CG_1^a	W 30x84	W 36x194	W 18x86	W 21x44	W 18x86
	CG_2	W 27x146	W 27x146	W 21x50	W 14x109	W 14x74
	CG_3	W 27x84	W 40x174	W 36x135	W 10x68	W 18x76
	CG_4	W 27x84	W 21x62	W 33x201	W 27x146	W 21x93
	CG_5	W 27x84	W 24x76	W 30x108	W 40x215	W 12x170
	IB^a	W 27x84	W 14x30	W 21x57	W 16x45	W 14x30
	OB^a	W 27x84	W 16x31	W 16x26	W 16x36	W 21x62
	BR^a	W 27x84	W 33x118	W 18x76	W 8x31	W 16x77
weight (ton)		634.12	612.05	584.93	**559.323**	**540.384**

^{a} CG denotes Column Group concerning Fig. 10, IB: Inner Beams, OB: Outer Beams, BR: Bracing

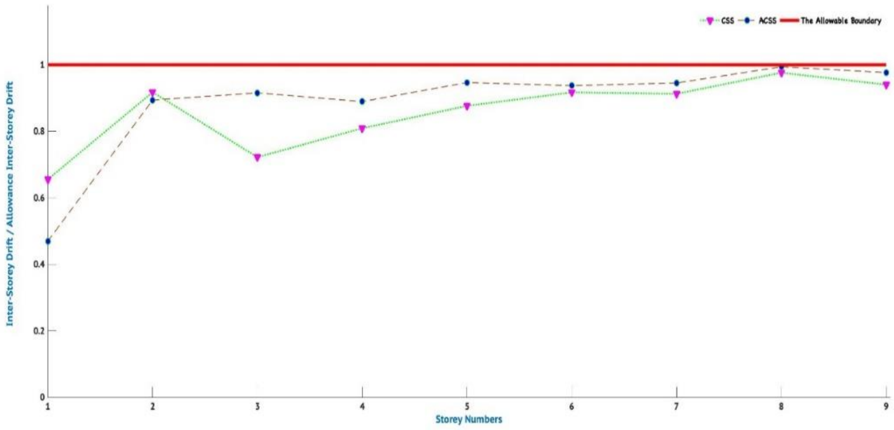


Figure 13. The ratio of the inter-story drift to the allowable inter-story drift in the optimum frame design for the 10-story with 1026-Member steel using the ACSS and CSS algorithms

5. CONCLUDING REMARKS

The main goal of the present work is to investigate the performance of the advanced CSS for optimum design of structures. The numerical results obtained through practical design optimization of four structures, clearly show that the ACSS is capable of reducing the weight of structures. The obtained designs reveal the efficiency and competitiveness of the ACSS compared to its standard version and different other algorithms in terms of the best optimal weight. It should be noted that the convergence speed of the proposed algorithm is better than the CSS. As future research, there are many works to be done, to name a few, the presented ACSS can be applied to other problems in engineering.

REFERENCES

1. Kaveh A, *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 2nd edition, Springer, Switzerland, 2017.
2. Talatahari S, Azizi MJC. Engineering, optimization of constrained mathematical and engineering design problems using chaos game optimization, 2020: p. 106560. https://doi.org/10.1016/j.cie.2020.106560.
3. Hasançebi O, Çarbaş S, Doğan E, Erdal F, Saka MP. Comparison of non-deterministic search techniques in the optimum design of real size steel frames, *Comput Struct* 2010; **88**: 1033-1048. https://doi.org/10.1016/j.compstruc.2010.06.006.
4. Camp CV, Bichon BJ, Stovall SP. Design of steel frames using ant colony optimization, *J Struct Eng* 2004; **131**: 369-79. https://doi.org/10.1061/(ASCE)0733-9445(2005)131:3(369).
5. Kaveh A, Abbasgholiha H. Optimum design of steel sway frames using big bang-big

crunch algorithm, *Asian J Civil Eng* 2011; **12**: 293-317.

6. Perez RE, Behdinan K. Particle swarm approach for structural design optimization, *Comput Struct* 2007, **85**: 1579-88. https://doi.org/10.1016/j.compstruc.2006.10.013.
7. Carbas S. Design optimization of steel frames using an enhanced firefly algorithm, *Eng Optim* 2016; **48**: 2007-25. https://doi.org/10.1080/0305215X.2016.1145217.
8. Kaveh A, Talatahari S, Khodadadi N. The hybrid invasive weed optimization-shuffled frog-leaping algorithm applied to optimal design of frame structures, *Period Polytech Civil Eng* 2019, **63**: 882-97. https://doi.org/10.3311/PPci.14576.
9. Saka MP. Optimum design of steel sway frames to bs5950 using the harmony search algorithm, *J Construct Steel Res* 2009; **65**: 36-43. https://doi.org/10.1016/j.jcsr.2008.02.005.
10. Kaveh A, Bolandgerami A. Optimal design of large-scale space steel frames using cascade enhanced colliding body optimization, *Struct Multidisci Optim* 2017; **55**: 237-56. https://doi.org/10.1007/s00158-016-1494-2.
11. Hasançebi O, Carbas S. Bat inspired algorithm for discrete size optimization of steel frames, *Adv Eng Softw* 2014; **67**: 173-185. https://doi.org/10.1016/j.advengsoft.2013.10.003.
12. Maheri MR, Narimani M. An enhanced harmony search algorithm for optimum design of side sway steel frames, *Comput Struct* 2014; **136**: 78-89. https://doi.org/10.1016/j.compstruc.2014.02.001.
13. Alberdi R, Khandelwal K. Comparison of robustness of metaheuristic algorithms for steel frame optimization, *Eng Struct* 2015; **102**: 40-60. https://doi.org/10.1016/j.engstruct.2015.08.012.
14. Bybordiani M, Kazemzadeh Azad S. Optimum design of steel braced frames considering dynamic soil-structure interaction, *Struct Multidisc Optimization* 2019; **60**: 1023-1037. https://doi.org/10.1007/s00158-019-02260-4.
15. Kaveh A, Talatahari S. Charged system search for optimal design of frame structures, *Appl Soft Comput* 2012; **12**: 382-393. https://doi.org/10.1016/j.asoc.2011.08.034.
16. Zakian P, Kaveh A. Economic dispatch of power systems using an adaptive charged system, *Appl Soft Comput* 2018; **73**: 607-22. https://doi.org/10.1016/j.asoc.2018.09.008.
17. Kaveh A, Khodadadi N, Farahamand Azar B, Talatahari S. Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections, *Eng Comput* 2020: 1435-5663. https://doi.org/10.1007/s00366-020-00955-7.
18. Kazemzadeh Azad S, Hasançebi O, Kazemzadeh Azad S. Upper bound strategy for metaheuristic based design optimization of steel frames, *Adv Eng Softw* 2013; **57**: 19-32. https://doi.org/10.1016/j.advengsoft.2012.11.016.
19. Hasançebi O, Kazemzadeh Azad S. Discrete size optimization of steel trusses using a refined big bang–big Crunch Algorithm, *Eng Optim* 2014; **46**: 61-83. https://doi.org/10.1080/0305215X.2012.748047.
20. Hasançebi O, Kazemzadeh Azad S. An exponential big bang–big crunch algorithm for discrete design optimization of steel frames, *Comput Struct* 2012; **110-111**: 167-79. https://doi.org/10.1016/j.compstruc.2012.07.014.
21. Tizhoosh HR, Ventresca M, Rahnamayan SH. *Opposition-Based Computing, in Oppositional Concepts in Computational Intelligence*, Berlin, Springer Berlin

Heidelberg, 2008: pp. 11-28.

22. Haklı H, Uğuz H. A novel particle swarm optimization algorithm with levy flight, *Appl Soft Comput* 2014; **23**: 333-45. https://doi.org/10.1016/j.asoc.2014.06.034.
23. Yang XS. *Cuckoo Search and Firefly Algorithm, in Theory and Applications*, Springer International Publishing, 2013.
24. American Institute of Steel Construction (AISC), in Manual of Steel Construction-Load Resistance Factor Design, Chicago, AISC, 2001.
25. McGuire W, Steel Structures, Englewood Cliffs NJ, Prentice Hall, 1968.
26. Kaveh A, Talatahari S. A novel heuristic optimization method: Charged system search, *Acta Mech* 2010; **213**: pp. 267-286. https://doi.org/10.1007/s00707-009-0270-4.
27. Barisal AK, Prusty RC. Large scale economic dispatch of power systems using oppositional invasive weed optimization, *Appl Soft Comput* 2015; **29**: 122-37. https://doi.org/10.1016/j.asoc.2014.12.014.
28. Kaveh A, Talataharai S. An enhanced charged system search for configuration optimization using the concept of fields of forces, *Struct Multidisc Optim* 2011; **43**: 339-51. https://doi.org/10.1007/s00158-010-0571-1.
29. Kaveh A, Motie Share M, Moslehi M. A new meta-heuristic algorithm for optimization: magnetic charged system search, *Acta Mech* 2013; **224**: 85-107. https://doi.org/10.1007/s00707-012-0745-6.
30. Kaveh A, Talataharai S. A hybrid CSS and PSO algorithm for optimal design of structures, *Struct Eng Mech* 2012; **42**: 783-97. http://dx.doi.org/10.12989/sem.2012.42.6.783.
31. Kaveh A, Talataharai S. Size optimization of space trusses using Big Bang–Big Crunch algorithm, *Comput Struct* 2009; **87**: 1129-40. https://doi.org/10.1016/j.compstruc.2009.04.011.
32. Kaveh A, Talataharai S. Charged system search for optimum grillage system design using the LRFD-AISC code, *J Construct Steel Res* 2010; **66**: 767-71. https://doi.org/10.1016/j.jcsr.2010.01.007.