Decision Analytics Journal 3 (2022) 100067

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

# Optimal design of low- and high-rise building structures by Tribe-Harmony Search algorithm

Mahdi Azizi [a,*], Siamak Talatahari [b], Mahla Basiri [a], Milad Baghalzadeh Shishehgarkhaneh [c]

[a] Department of Civil Engineering, University of Tabriz, Tabriz, Iran
[b] Data Science Institute, Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia
[c] Department of Civil Engineering, Islamic Azad University of Tabriz, Tabriz, Iran

ARTICLE INFO

*Keywords:*
Optimum design
Building structure
Tribe-Harmony Search algorithm
Metaheuristic
Optimization

ABSTRACT

In this paper, optimum design of building structures is conducted by metaheuristic algorithms due to the shortcomings of the conventional design methods in providing economical designs. The Harmony Search (HS) is utilized as the main algorithm which was developed based on the musical process of searching for the optimal condition of harmony to produce an appropriate search approach for design optimization purposes. Besides, the Tribe-Harmony Search (Tribe-HS) algorithm is also proposed for the first time in this paper to improve the performance of the HS algorithm which divides the HS's searching phase into three distinct phases called "tribes", lead the primary algorithm to prioritize global search in the early iterations while resolving local search in the later iterations. Three building structures with 135, 3860, and 8272 structural members are used as design examples to demonstrate the suggested method's capacity to solve challenging optimization problems. The recommended method's overall performance is compared to that of the conventional Harmony Search algorithm and ten alternative metaheuristic algorithms using a total of 30 independent runs in each instance for statistical reasons. The findings demonstrated that the suggested method outperformed the other metaheuristics for the study design instances.

## 1. Introduction

Structural optimization is the process of finding the best configurations of elements for a structural system with consideration of design constraints and a fully developed objective function. In most cases, the total construction cost of the structure is considered objective functions in which the topology, size and shape of the structural systems have the main role in this purpose. Design constraints are the other aspect of the structural optimization process which demonstrate the structural behavior, including the deformation, force, fatigue, and damping of structural members. Structural optimization considers these objective functions and design constraints to provide a better configuration of elements for a structural system. In other words, structural optimization is an intelligently developed design concept in which the optimal configuration of the structural components is considered by means of a fully-established optimization algorithm.

The process of finding an optimal configuration of elements for a structural system requires an optimization algorithm that should be capable of providing better results than the traditionally developed design approaches. In this regard, the metaheuristic-based optimization approaches could be considered as optimization algorithms that have been utilized for optimization purposes in different fields for several decades. The Genetic Algorithm (GA) [1], Differential Evolution (DE) [2], Ant Colony Optimization (ACO) [3], Particle Swarm Optimization (PSO) [4], Charged System Search (CSS) [5–7], Material Generation Algorithm (MGA) [8,9], and Chaos Game Optimization (CGO) [10,11] are some of these methods. Additionally, several of the aforementioned metaheuristic algorithms have been used in various engineering problems, yet none of them has ever proven to be extremely the optimal method. Regardless matter how strong the algorithms are, several improvements may be made to basic algorithms to provide more accurate results with less computing time. These advancements are intended to enhance current algorithms or hybridize two or three of them to achieve reasonable outcomes for objective functions.

Many of the significant optimization algorithm improvements include: the enhanced PSO introduced by Wang, et al. [12], improved ACO proposed by Kaveh and Talatahari [13], upgraded Whale Optimization Algorithm proposed by Azizi, et al. [14], hybrid GA-Imperialistic Competitive Algorithm developed by Fasahat and Payvandy [15] and hybrid Ant Lion Optimizer-Jaya approach presented by Azizi, et al. [16]. Meanwhile, some other developed approaches are mentioned in Refs. [10,17–24].

While designing diverse engineering structures, one of the most difficult challenges for structural engineers is optimizing the structure's

* Corresponding author.
*E-mail address:* mehdi.azizi@tabrizu.ac.ir (M. Azizi).

https://doi.org/10.1016/j.dajour.2022.100067
Received 1 April 2022; Received in revised form 2 May 2022; Accepted 6 May 2022
Available online 13 May 2022

weight, shape, cost of construction, topology, and manufacturing time while considering numerous constraints. These features are frequently incorporated in an optimization problem that considers the best design sections of structural elements for minimizing the structure's weight by considering inequality and equality constraints. Farshchin, et al. [25] discussed the optimum design of multiple steel frame structures using School Based Optimization (SBO) algorithm. Khodadadi, et al. [26] proposed the multi-objective version of a recently proposed metaheuristic algorithm called Crystal Structure Algorithm (CryStAl) for engineering optimization problems. Kaveh and BolandGerami [27] proposed an upgraded version of Colliding Body Optimization (CBO) algorithm called "Cascade Enhanced CBO" for large-scale steel space frames' optimization. Talatahari, et al. [28] combined eagle strategy with DE algorithm for design optimization of different frame structures with steel sections. Maheri, et al. [29] developed an improved version of the Honey Bee Mating Optimization (HBMO) algorithm for optimum element determination of side-sway structural frames with steel design sections. Kaveh and Ghazaan [30] discussed size optimization of skeletal steel structures while an improved Whale Optimization Algorithm (WOA) has been proposed for this purpose. Kazemzadeh Azad, et al. [31] utilized the Big Bang–Big Crunch (BB–BC) as the main optimization algorithm for optimum design of different frame structures, while the Upper Bound Strategy (UBS) is implemented for enhancing the computational complexity of the main algorithm. Kazemzadeh Azad, et al. [32] discussed the optimum design of steel frame structures by combining the BB–BC algorithm with UBS to reduce the number of structural analyses needed as much as possible throughout the optimization procedure. Hasançebi [33] utilized Evolutionary Strategy (ES) for economic design optimization of multiple frame structures. Tort, et al. [34] discussed the optimal design of towers in real-world engineering for lattice transmission by utilization of Simulated Annealing (SA). Furthermore, Kundu and Garg [35] introduced an efficient hybrid approach for solving several types of engineering design and numerical optimization problems called enhanced teaching–learning Harris hawks optimization (ITLHHO), which uses improved teaching–learning-based optimization. Kaveh and Vazirinia [36] introduced an improved sine cosine algorithm (USCA), which utilizes a harmony search-based operator to increase exploration while also dealing with changeable constraints, and saves the best answers in an archive. Brajević and Tuba [37] developed an upgraded version of artificial bee colony (UABC) approach for constrained optimization problems that improves the fine-tuning properties of the modification rate parameter and uses the ABC algorithm's modified scout bee phase. To increase firefly algorithm's efficiency in handling constrained engineering optimization problems, Brajević and Ignjatović [38] suggested an updated firefly algorithm (UFA). The suggested methodology employs a set of feasibility-based criteria to guide the search to the most feasible section of the search space, as well as an enhanced boundary constraint scheme and an equality constraint approach. Pathak and Srivastava [39] proposed a new bat algorithm that includes a cuckoo search and Sugeno inertia weight (UBCSIW). The bat algorithm, which can exploit optimum solutions in search space, is merged with cuckoo search, which can explore the best solution globally utilizing Levy flight in the search space, in the proposed UBCSIW algorithm.

The critical contribution of most of the research studies reviewed is developing a preferable design method for optimum frame structure element configuration. Due to the shortcomings of conventional approaches with computational complexity difficulties, the importance of offering a thoroughly defined optimization procedure is growing. Determining the appropriate search space is one of the most prominent phases of designing an optimum design strategy for structural optimization. From a structural standpoint, using trial and error to create building structures using available wide-flange sections (W-shaped sections) does not fulfill the affordable aspects of engineering projects. As a result, this research focuses on the optimum design of real-size steel building structures, where more optimum and practical design sections may be chosen to determine the search space and give an appropriate structural design configuration. The Harmony Search (HS) algorithm is chosen as the primary optimization algorithm suggested by Geem, et al. [40] and is based on the musical practice of attempting to achieve perfect harmony. Zhang and Geem [41] reviewed and focused on the historical development of Harmony Search (HS) algorithm structure instead of applications; they elucidated adaption of original operators of the basic harmony search, parameter adaption, hybrid methods, handling multi-objective optimization problems and constraint handling. There has been a growing interest in enhancing the overall efficiency of this algorithm as a result of its many applications in various optimization domains [42–48]. Numerous improved, modified, or hybridized variants of the HS algorithm have been suggested and used for engineering design optimization as search techniques. Keshtegar, et al. [49] proposed a modified version of harmony search (HS) algorithm for improvisation procedure. Ouyang, et al. [50] suggested an improved version of HS algorithm for problems related to engineering design considering the general iteration models. Moon, et al. [51] proposed a novel approach to estimate the vanishing point using a harmony search (HS) algorithm; they claimed that HS stably estimates vanishing points with respect to statistics when compared with RANSAC. Yi, et al. [52] discussed the engineering design of different optimization problems by using parallel chaotic local search improved HS algorithm. Keshtegar, et al. [53] used a dynamic harmony search (DHS) algorithm for accurate calibration of strength and strain enhancement ratios of FRP-confined concrete. Hasanipanah, et al. [54] proposed an ANN-adaptive dynamical harmony search algorithm for accurate prediction of blast-induced flyrock. Yi, et al. [55] developed an improved HS algorithm considering a multi-level screening strategy for design optimization of engineering problems. Sheikholeslami, et al. [56] discussed the optimum design of water distribution systems utilizing a hybrid optimization method developed based on cuckoo search (CS) and HS algorithms. Keshtegar, et al. [57] proposed a bi-loop optimization framework of stiffened panels is proposed to search the global optimum, including an adaptive response surface method (ARSM) loop and a Gaussian global-best harmony search (GGHS) loop. Ouyang, et al. [58] proposed a hybrid metaheuristic approach by utilizing Teaching–Learning Based Optimization (TLBO) and HS algorithms for optimum design of difficult problems in engineering. Keshtegar and Etedali [59] proposed based on the dynamical parameters that are adjusted using the previous results of the harmony memory with a simple formulation. Gholizadeh and Barzegar [60] developed a sequential HS algorithm for shape optimization of different structures by considering frequency constraints. Jaberipour and Khorram [61] discussed mixed–discrete problems in the engineering optimization field by utilizing an enhanced HS algorithm. This study proposes the Tribe-Harmony Search (Tribe-HS) algorithm, in which the primary notion of improvement is derived from the "Tribe-CSS" method proposed by Talatahari and Azizi [21]. These phases considered tribes lead the algorithm to concentrate on global searching in the early iterations while local searching is handled in the later iterations in the Tribe-HS technique. These adjustments improve the exploration and exploitation rates of the standard algorithm. To assess the suggested method's ability to deal with complex optimization problems, three different building structures with 3, 20, and 60 stories with 135, 3860, and 8272 structural members are deemed as design examples. The W-shaped design sections for structural components in these structures are utilized to analyze the design requirements, and the AISC-LRFD [62] code for steel structure design is applied. The suggested method's overall performance is compared to that of the conventional HS algorithm and several metaheuristics, with 30 independent runs performed in each example for statistical reasons.

The remainder of the paper is divided into the following sections. In Section 2, the optimum design of steel frames, including objective function and design constraints are presented. Section 3 describes the utilized optimization algorithm in detail. In Sections 4 and 5, design examples, including 3, 20, and 60-story steel structures, alongside alternative metaheuristic algorithms are illustrated, respectively. Numerical results have been reported in Section 6. Finally in Section 7, the core findings of this study are presented as concluding remarks.

## 2. Optimum design of steel frames

### 2.1. Objective function

There is an assumption in the optimum design of steel structural frames that $N_m$ structural members are classified to $N_d$ design groups. To reduce the total weight of the structure, the sequence numbers in steel design sections given to $N_d$ member groups are calculated using a vector of integer values. The integer vector and total weight of the analyzed structure are summarized below.

$$\text{Find} \quad I^T = \left[I_1, I_2, \dots, I_{N_d}\right] \tag{1}$$

$$\text{To minimize} \quad W = \sum_{i=1}^{N_d} \rho_i . A_i \sum_{j=1}^{N_t} L_j \tag{2}$$

where, $\rho_i$ and $A_i$ are the steel design section's unit weight and length established for member group *i*, respectively; $L_j$ shows the length of the *jth* member associated with the *ith* group, and $N_t$ indicates the overall number of all structural members in group $i$.

### 2.2. Design constraints

The AISC-LRFD [29] code for steel structure design specifies two primary design criteria: strength and serviceability. When attempting to minimize the structures' weight, for the design sections' strength criteria, the following constraints must be met:

$$C_{IEL}^{i} = \left[\frac{P_{uJ}}{\varphi P_n}\right]_{IEL} + \frac{8}{9}\left(\frac{M_{uxJ}}{\varphi_b M_{nx}} + \frac{M_{uyJ}}{\varphi_b M_{ny}}\right)_{IEL} - 1 \le 0 \quad for \quad \left[\frac{P_{uJ}}{\varphi P_n}\right]_{IEL} \ge 0.2 \tag{3}$$

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

$$C_{IEL}^{v} = \left(V_{uJ}\right)_{IEL} + \left(\varphi_v V_n\right)_{IEL} \le 0 \tag{5}$$

where, $IEL$ shows the element number as $IEL = 1, 2, \dots, NEL$ and $NEL$ is the total number of elements; $J$ indicates the number of load combination as $J = 1, 2, \dots, N$ and $N$ is the overall number of whole design load combinations; $P_{uJ}$ indicates the compressive or tensile (axial) strength that is needed for the $J$th design load; $M_{uxJ}$ and $M_{uyJ}$ are the total flexural strengths needed for bending of structural elements concerning $x$ and $y$, under the $J$th design load combination, respectively; the $x$ and $y$ subscripts are used as related symbols for strong and weak axes bending, respectively. $P_n$, $M_{nx}$ and $M_{ny}$ are the nominal compressive or tensile (axial) and flexural (for bending of structural elements about $x$ and $y$ axes) strengths of the $IEL$th member under consideration. $\varphi$ clarifies the axial strength resistance factor formulated about the gross section yielding which for compression and tension are 0.85 and 0.9, respectively. $\varphi_b$ shows the flexural resistance factor (0.9). The shear strength needed under the *Jth* design load combination is denoted by $V_{uJ}$, and $V_n$ elucidates the nominal shear strength of the $IEL$th deemed elements and $\varphi_v$ equals 0.9.

## 3. Utilized optimization algorithms

This part discusses the metaheuristic optimization algorithms that were used, including the conventional Harmony Search method and its improved variant, named "Tribe-HS".

### 3.1. Harmony search algorithm

The fundamental concept behind the development of a novel optimization method called "Harmony Search" is that in a musical performance process, a musician naturally conducts a proper searching process to discover a better state of harmony with multiple tries. In jazz improvisation, the player tends to achieve a musically pleasant harmony as a perfect state by considering the aesthetic aspects. This procedure is analogous to the optimization process, in which the optimization algorithm strives to attain the global solution as a perfect state by taking the objective function evaluation into account. Each musical instrument's pitch controls the aesthetic aspect of musical performance as the values of decision variables control the objective function evaluation. The HS algorithm's mathematical formulation is constructed in five major phases, each of which is discussed in detail.

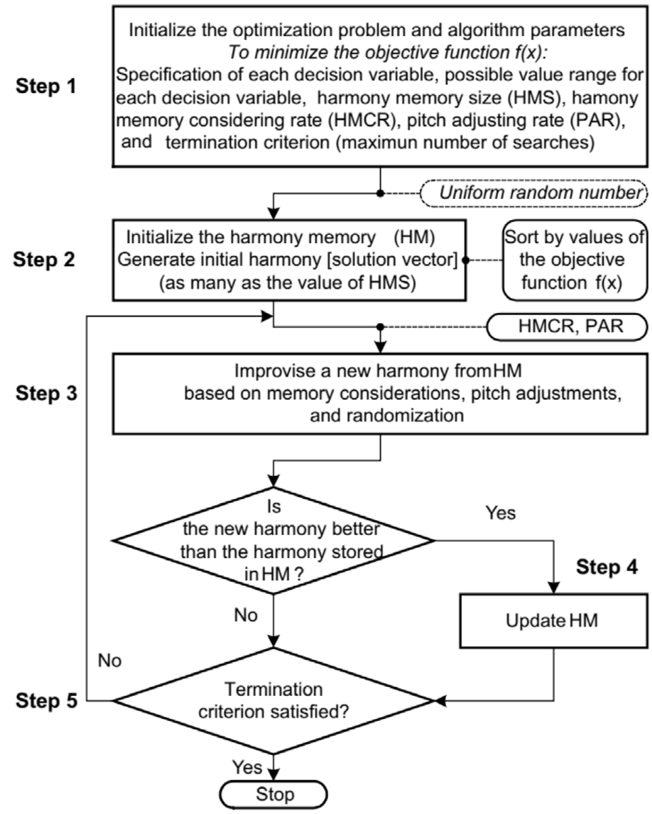


**Fig. 1.** Flowchart of the HS algorithm [63].

The initialization procedure is carried out in the first phase, in which the initial values for the harmony vectors ($X_i$) consisting of different decision variables ($X_i = \{x_1, x_2, \dots, x_i, \dots, x_n\}$) and their related objective function amounts ($F_i$) are determined. The decision variables demonstrate different musicians, and the objective function evaluations demonstrate the harmony which these musicians achieve. In this step, the crucial parameters of the HS algorithm such as the Harmony Memory Size (HMS), Pitch Adjusting Rate (PAR), Harmony Memory Considering Rate (HMCR), and the termination criteria, which is deemed as the maximum number of iterations (MaxIter) are determined. The PAR and HMCR parameters are utilized to improve each solution vector's quality in the optimization process.

The initial Harmony Memory (HM) is determined in the second phase, including the solution vectors generated randomly with the harmony memory (HMS) size, classified regarding their objective function's values. The mathematical presentation of the HM is as follows:

$$HM = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{HMS} \end{bmatrix} \tag{6}$$

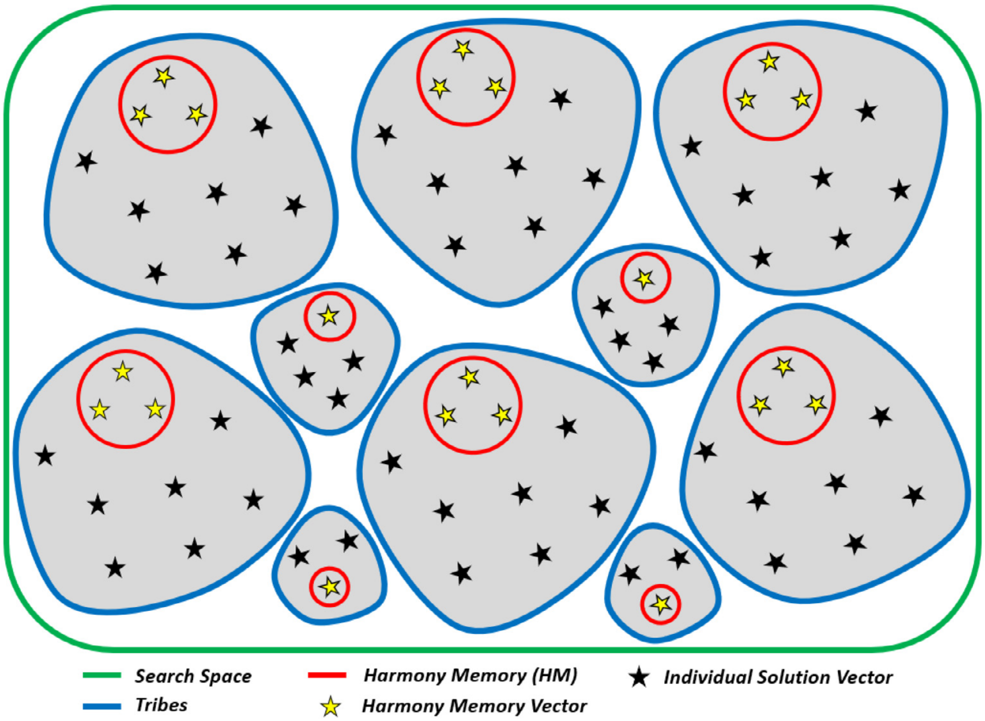


**Fig. 2.** Schematic demonstration of the search procedure for the Tribe-HS in the first phase.

In step three, a new harmony vector ($X_i' = \{x_1', x_2', \dots, x_i', \dots, x_n'\}$) is improvised from the harmony memory or initial harmony vectors based on the pitch adjustment, memory consideration, and randomization process. The decision variables can be determined by choosing any values from the HM in Eq. (6) or choosing from the initial harmony vectors. In this regard, a random number distributed uniformly in the range of (0, 1) is produced to decide between two choices. If the produced random number is higher than the previously determined HMCR, the novel harmony vector is selected from the HM, while for the random numbers lower than the HMCR, the novel vector is determined to form the initial harmony vectors ($X_i$). These aspects are mathematically represented as follows:

$$x_i' \rightarrow \begin{cases} x_i' \in HM & with\ the\ probability\ of\ \ (HMCR) \\ x_i' \in X_i & with\ the\ probability\ of\ \ (1-HMCR) \end{cases} \tag{7}$$

Pitch adjustment is used to mathematically model the mutation phase of the procedure for the values obtained from the HM by creating another random value spread equally within the range of (0, 1). If the created random number is more than the previously determined PAR, the novel harmony vector selected from the HM will choose a neighboring value with the PAR probability; however, no pitch adjustment is made if the generated random number is less than the PAR. These considerations are mathematically expressed as follows:

$$x_i' \rightarrow \begin{cases} x_i' + (bw \times u) & with\ the\ probability\ of\ \ (PAR) \\ x_i' & with\ the\ probability\ of\ \ (1-PAR) \end{cases} \tag{8}$$

where $bw$ shows an arbitrary distance bandwidth and $u$ is a uniformly distributed random number in the range of (−1, 1).

In step four, the HM is upgraded, and if the newly created harmony vector outperforms the worst harmony in the HM concerning the value of the objective function, the novel harmony is replaced by the worst, and the HM is sorted using the objective function values. The third and fourth phases are repeated in the fifth step until the termination requirements are met. The flowchart of the suggested HS algorithm is demonstrated in Fig. 1.

### 3.2. Tribe-harmony search algorithm

Premature convergence is a possibility for a large number of optimization algorithms. An intriguing endeavor has been made to strengthen the general capacity of metaheuristic approaches and algorithms by providing appropriate solutions to the algorithms' inadequacies in the past few decades. In this respect, this study proposed the notion of Tribe-HS to improve the HS algorithm's potential of solving challenging optimization problems. This idea relies on the fact that by separating the search space into many separate groupings known as "Tribes", the searching process is carried out in an old-fashioned manner in which the tribes may offer a civilized way of life without connecting in the early ages. Nonetheless, These tribes try to exchange information and, in later years, even unite for a better way of life. The search area in the HS algorithm is separated into many tribes (search spaces) based on the presented notion, with each tribe's searching procedure completed in a unique way that increases the standard algorithm's performance.

To mathematically represent the above notion, a maximum number of tribes ($N_t$) should be established, which will be used to divide the solution vectors in the search space into these tribes. Each of the aforementioned tribes has a random number of solution vectors ($N_s$), and the searching process is carried out in these tribes in a particular manner to converge on a correct solution effectively. The algorithm's primary search phase is split into three distinct stages: the isolated, the communing, and the unified phase. The new formulation includes the stopping requirements, which divides the maximum number of function evaluations or iterations into three parts.

The isolated phase of the algorithm is the initial phase in which the solution vectors in the search space and inside the HM are not allowed to exchange information or personal experiences with other tribes. This procedure is repeated until the predetermined stopping conditions, which are separated into the three stages indicated above, are met. The second step, called the communing phase, allows tribes to utilize the solution vectors in each other's HMs and update their

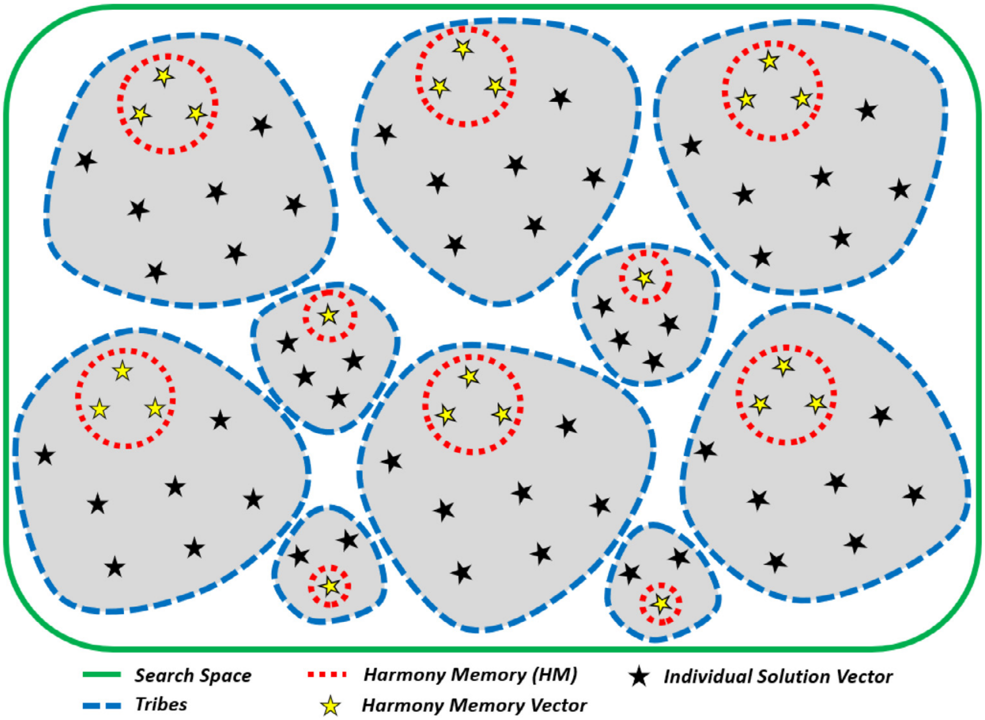


**Fig. 3.** Schematic demonstration for the Tribe-HS in the second phase in the search procedure.

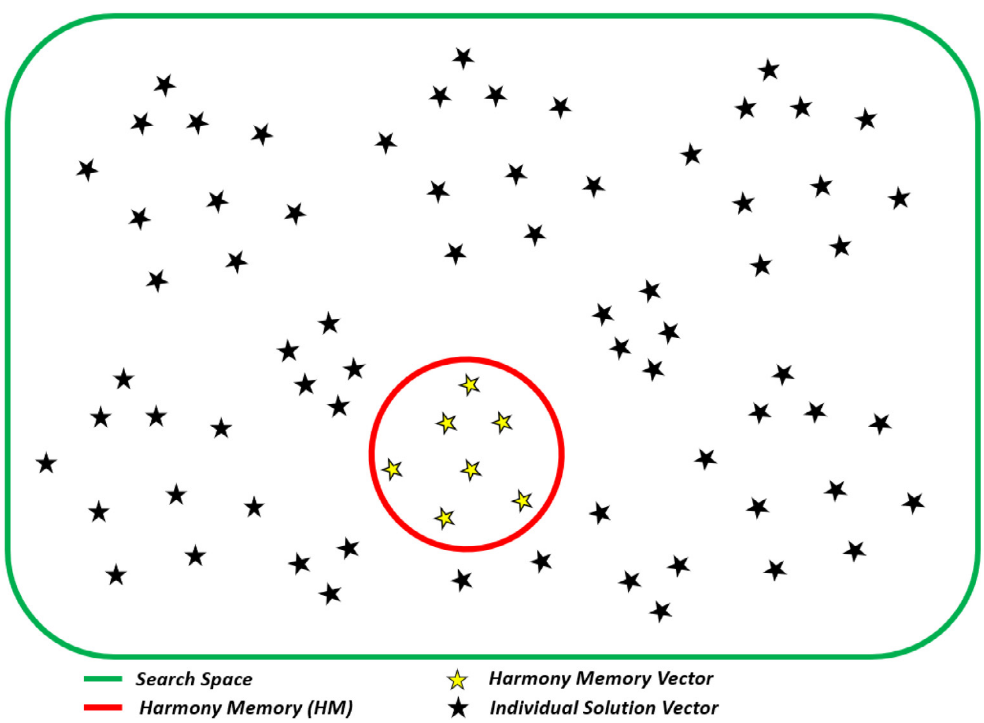


**Fig. 4.** Schematic demonstration of the search procedure for the Tribe-HS in the third Phase.

most recently discovered information. As the third phase, the unified phase brings together all of the solution vectors from separate tribes, and this phase continues until the stated termination requirement is fulfilled. The schematic representation of the Tribe-HS algorithm in its three distinct stages is shown in Figs. 2–4, while the pseudo-code for this algorithm is shown in Fig. 5.

## 4. Design examples

This part contains detailed information about the 3 real-size steel building structures that are used to assess the Tribe-HS's capability to evaluate the structural elements' optimal design sections. Different plans select these structures and in diverse heights to find out the

```
Procedure: Tribe-Harmony Search Algorithm (Tribe-HS)
Initialize the harmony vectors Xi (i=1, 2, …, NH)
Evaluate the fitness of each harmony vector
Define HMCR
Define PAR
Generate HM
Define the number of considered tribes (NT)
Define maximum number of iterations for each phase (T1, T2, T3)
  While (It<Maximum number of iterations)
    Phase I
    for It=1:T1
      for t=1 to NT
        for i=1:NH
          if rand<HMCR
            Select a harmony vector form the HM
              if rand<PAR
                Adjust the selected harmony vector
                Evaluate the fitness of the harmony vector
              end
          else
            Select a random harmony vector from search space
            Evaluate the fitness of the harmony vector
          end
        end
        Update the HM
      end
    end
    Phase II
    for It= T1+1:T2
      for t=1 to NT
        for i=1:NH
          if rand<HMCR
            Select a harmony vector form the HM
              if rand<PAR
                Adjust the selected harmony vector
                Evaluate the fitness of the harmony vector
              end
          else
            Select a random harmony vector from search space
            Evaluate the fitness of the harmony vector
          end
        end
        Update the HM
      end
      Share the HM of tribes
    end
    Phase III
    for It= T2+1:T3 (It)
      for i=1:NH
        if rand<HMCR
          Select a harmony vector form the HM
            if rand<PAR
              Adjust the selected harmony vector
              Evaluate the fitness of the harmony vector
            end
        else
          Select a random harmony vector from search space
          Evaluate the fitness of the harmony vector
        end
      end
      Update the HM
    end
  end while
end procedure
```

**Fig. 5.** Pseudocode of the Tribe-HS optimization algorithm.

usefulness of the improved optimization algorithm in dealing with different sorts of building structures. The material characteristics used in these structures are stainless steel with an elasticity modulus (E) of 200 GPa, yield stress (Fy) of 248.2 MPa, and steel unit weight (q) of 7.85 ton/m$^3$.

To assist in the design process, the deemed building structures are exposed to ten different load combinations, as listed in Table 1. On typical floor beams, the acting dead and live loads are 14 and 10 kN/m, respectively, while the dead and live loads on roof beams are 12 and 7 kN/m, respectively. The seismic and wind loads on the

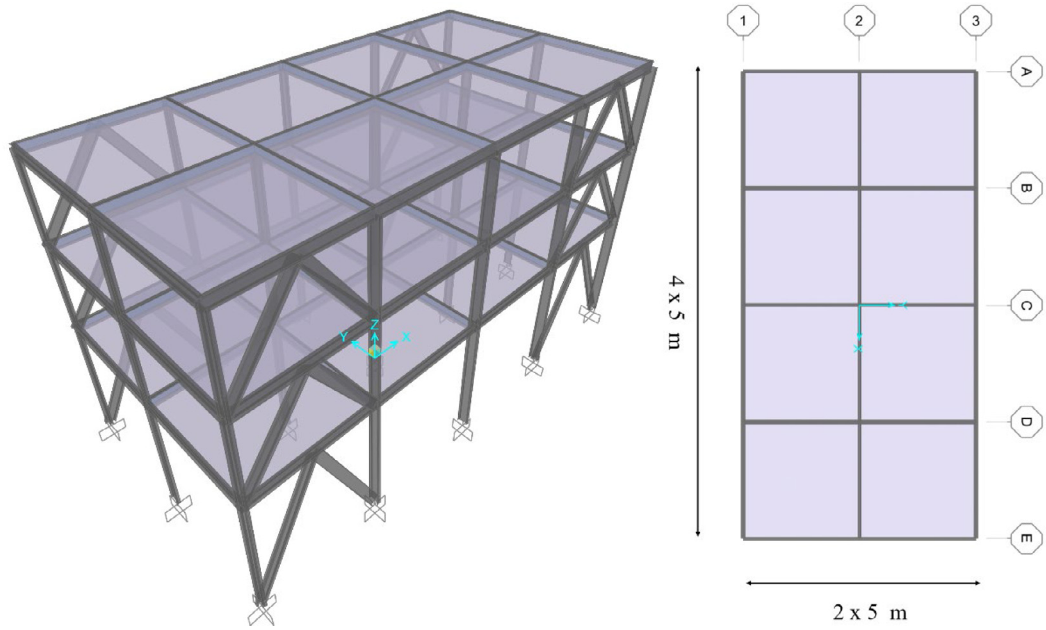


**Fig. 6.** The schematic representation and plan views of the 3-story steel structure.

**Table 1**
Load combinations for steel structural design.

| No. | Combination |
|---|---|
| 1 | 1.4 D |
| 2 | 1.2 D + 1.6 L |
| 3 | 1.2 D + 1.0 ($E_x$/$W_x$) + 0.5 L |
| 4 | 1.2 D + 1.0 ($E_{ex}$/$W_{ex}$) + 0.5 L |
| 5 | 1.2 D + 1.0 ($E_y$/$W_y$) + 0.5 L |
| 6 | 1.2 D + 1.0 ($E_{ey}$/$W_{ey}$) + 0.5 L |
| 7 | 0.9 D + 1.0 ($E_x$/$W_x$) |
| 8 | 0.9 D + 1.0 ($E_{ex}$/$W_{ex}$) |
| 9 | 0.9 D + 1.0 ($E_y$/$W_y$) |
| 10 | 0.9 D + 1.0 ($E_{ey}$/$W_{ey}$) |

D: Dead Load, L: Live Load, E: Earthquake Load, W: Wind Load.
$x$ and $y$: Loading directions without eccentricity.
$ex$ and $ey$: Loading directions with eccentricity.

structural systems under consideration are assessed in accordance with ASCE [64], establishing the minimum design loads for buildings and other structures.

### 4.1. Example 1: 3-story, 135-member steel structure

A three-story steel structure with 135 structural members is the first design example. The structural members of this structure are composed of 45 column elements, 66 beam elements, and 24 brace elements deemed as standard W-shaped sections. The moment resisting connections alongside inverted V-bracings are utilized as a lateral resisting system of the structure. The schematic and plan views of this structure are shown in Fig. 6, while the elevation views are shown in Fig. 7.

All 135 structural components of the three-story steel structure are classified into ten member groups based on their practical fabrication requirements. Member grouping is evaluated at both the plan and elevation levels, whereas structural members at the elevation level are grouped within each story. Each story's beams and braces are regarded to be part of a single beam and bracing group, whereas the columns are divided into four distinct groups at the plan level, as seen in Fig. 8.

### 4.2. Example 2: 20-story, 3860-member steel structure

The second design problem is the structural design of a 20-story steel structure with 3860 structural components. This design sample has 1064 columns, 1836 beams, and 960 bracing parts, with the columns, beams, and braces all having standard W-shaped design sections. This structure's lateral resistance is provided by cross-bracing systems in the X and Y axes, in addition to moment-resisting connections. Fig. 9 depicts the schematic and plan views of this structure.

Based on their fabrication requirements, all 3860 structural members of the 20-story steel structure are divided into 73 member groups. Member grouping is evaluated at both the plan and elevation levels, with structural members at the elevation level grouped every two stories. Additionally, the columns are classified into five distinct groups at the plan level, as illustrated in Fig. 10. Two groups are examined for beams: inner and outer beams, while one group is selected for each of the structure's neighboring two stories. As a result, 43 column design groups, 20 beam design groups, and ten bracing design groups are examined concerning the structure's plan and elevation levels.

### 4.3. Example 3: 60-story, 8272-member steel structure

The third design example is a 60-story steel structure with a structural tube system comprised of 8272 structural components. 3960 columns, 3960 beams, and 352 bracing components are evaluated in this design example, in which the design sections for the beams, columns, and braces are considered as standard W-shaped sections. The mega-bracing systems with the X and Y directions, as well as the moment-resisting connections, are used to prepare the lateral stability of this structure for the first 24 stories, while the lateral resisting system is used as a standard bracing system alongside the moment-resisting connections for the 25th to 60th stories. Fig. 11 shows a schematic representation of this structure.

Regarding the practical fabrication requirements, all 8272 members of the studied 60-story steel structure are divided into 103 member groups. The member grouping process is performed at both the elevation and plan levels, with the structural member grouping process occurring every six stories at the elevation level. Two column groups, the corner, and side column groups, are considered for tubes A to D in plan levels (Fig. 12), while each tube has one beam group. Every six stories, the bracing components are considered a separate group.

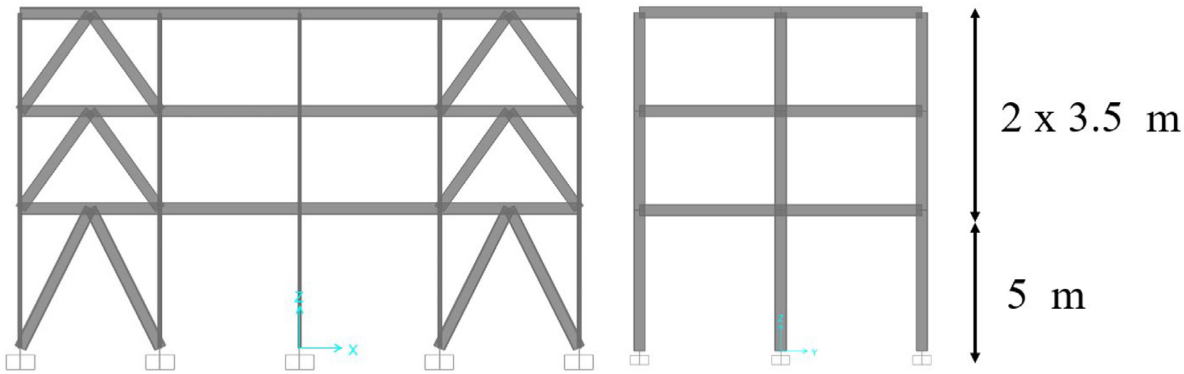


**Fig. 7.** The elevation views of the 3-story steel structure in $X$ and $Y$ directions.

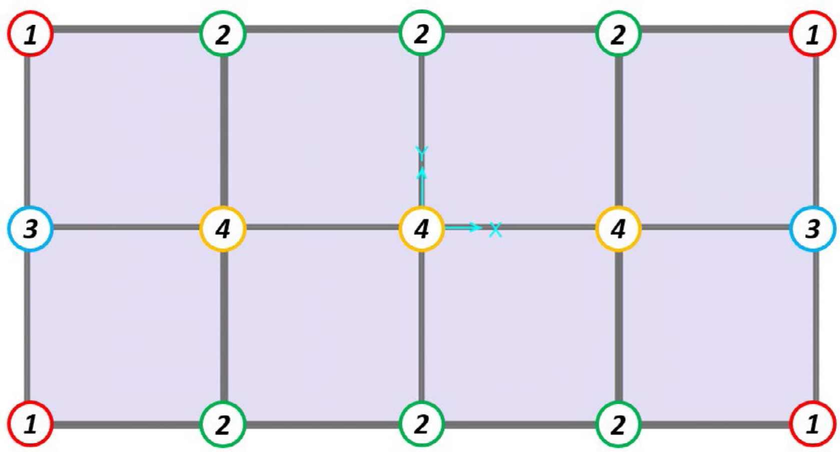


**Fig. 8.** Column grouping of the 3-story steel structure.

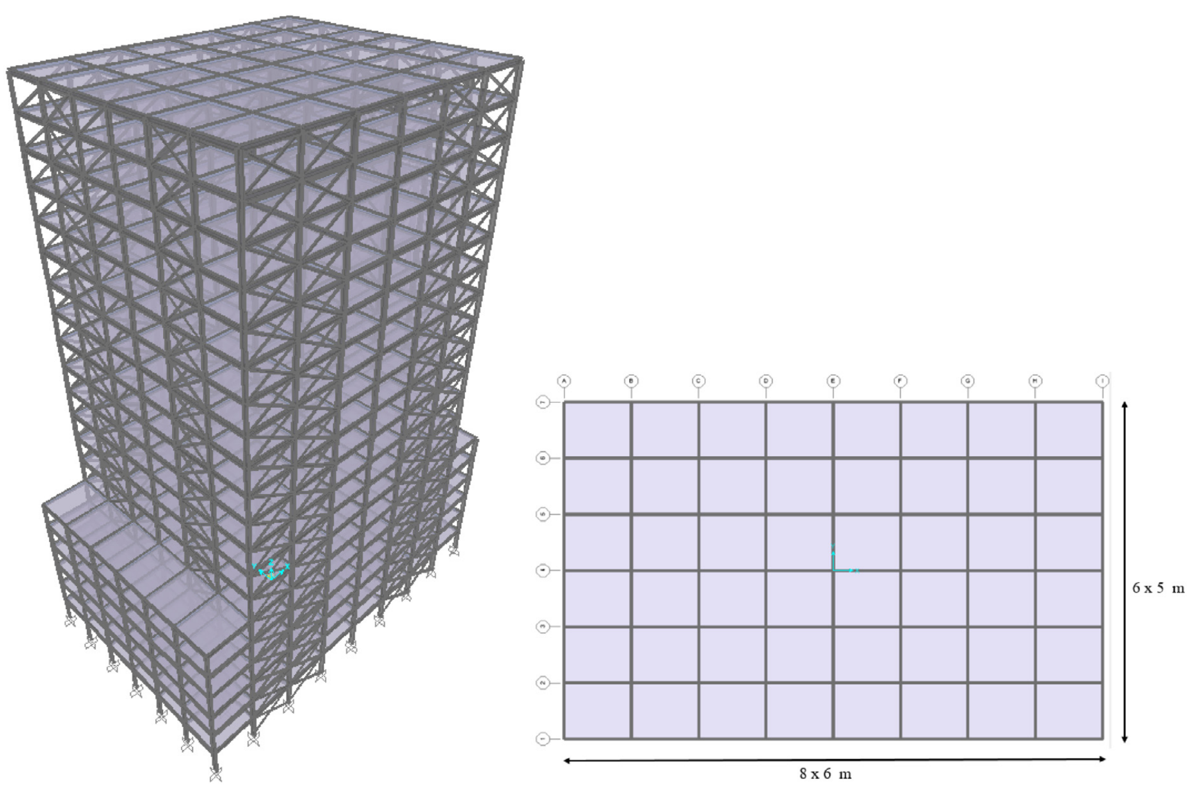


**Fig. 9.** The schematic and plan views of the 20-story steel structure with 3860 members.

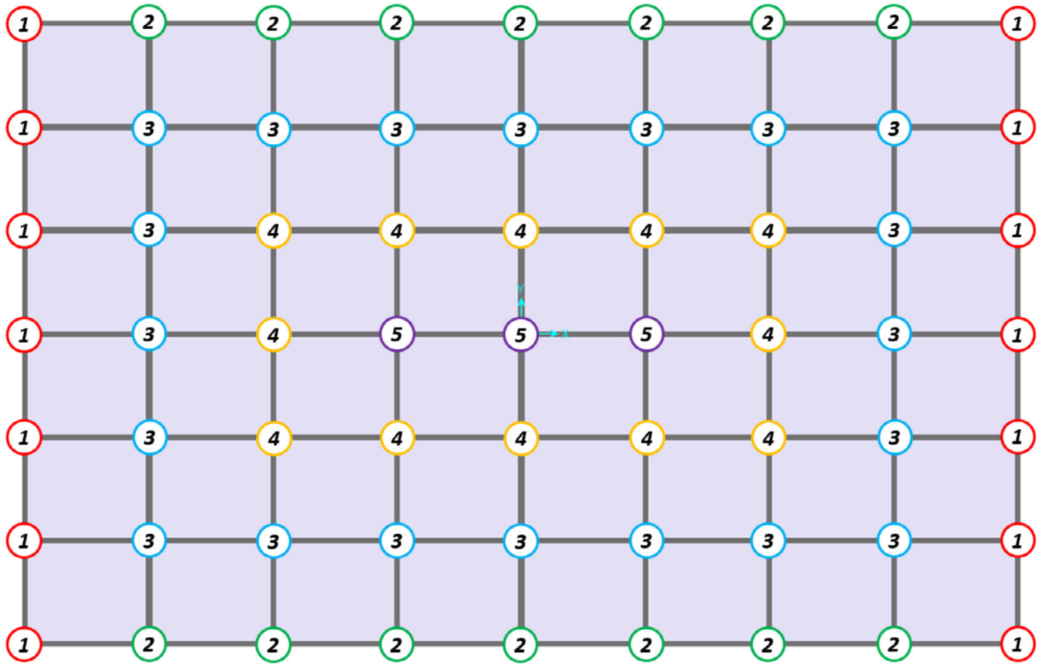


**Fig. 10.** Column grouping of the 20-story steel structure with 3860 members.

**Table 2**
Internal parameters for the alternative metaheuristic algorithms.

| Metaheuristic | Parameter | Description | Value |
|---|---|---|---|
| GA | $p_c$ | Crossover percentage | 0.8 |
| | $p_m$ | Mutation percentage | 0.3 |
| | $\mu$ | Mutation rate | 0.02 |
| | $\beta$ | Roulette wheel selection pressure | 1 |
| ACO | $N_s$ | Sample size | 50 |
| | $q$ | Intensification factor | 0.5 |
| | $\zeta$ | Deviation-distance ratio | 1 |
| PSO | $w$ | Inertia weight | 1 |
| | $w_d$ | Inertia weight damping ratio | 0.99 |
| | $c_1$ | Personal learning coefficient | 2 |
| | $c_2$ | Global learning coefficient | 2 |
| ICA | $N_{emp}$ | Number of empires/imperialists | 10 |
| | $\alpha$ | Selection pressure | 1 |
| | $\beta$ | Assimilation coefficient | 1.5 |
| | $p_r$ | Revolution probability | 0.05 |
| | $\mu$ | Revolution rate | 0.1 |
| | $\zeta$ | Colonies mean cost coefficient | 0.2 |
| BOA | $p$ | Probability switch | 0.8 |
| | $pe$ | Power exponent | 0.1 |
| | $sm$ | Sensory modality | 0.01 |
| KH | $V_f$ | Foraging speed | 0.02 |
| | $D_{max}$ | Maximum diffusion speed | 0.005 |
| | $N_{max}$ | Maximum induced speed | 0.01 |
| CSS | $a$ | Radius of charged sphere | 0.1 |
| | $HMCR$ | Harmony memory consideration rate | 0.85 |
| | $PAR$ | Pitch adjustment rate | 0.15 |
| | $kt$ | Attract-repel coefficient | 0.9 |
| | $N_{cm}$ | Charged memory size | 12 |
| | $k_a$ | Acceleration coefficient | 0.5 |
| | $k_v$ | Velocity coefficient | 0.5 |
| HS | $HMS$ | Harmony memory size | 50 |
| | $N_{new}$ | Number of new harmonies | 20 |
| | $HMCR$ | Harmony memory consideration rate | 0.9 |
| | $PAR$ | Pitch adjustment rate | 0.1 |
| | $FW$ | Fret width (bandwidth) | ±0.02 |
| | $FW_{damp}$ | Fret width damp ratio | 0.995 |

**Table 2** (*continued*).

| Metaheuristic | Parameter | Description | Value |
|---|---|---|---|
| Tribe-HS | $HMS$ | Harmony memory size | 50 |
| | $N_{new}$ | Number of new harmonies | 20 |
| | $HMCR$ | Harmony memory consideration rate | 0.9 |
| | $PAR$ | Pitch adjustment rate | 0.1 |
| | $FW$ | Fret width (bandwidth) | ±0.02 |
| | $FW_{damp}$ | Fret width damp ratio | 0.995 |
| | $N_T$ | Number of considered tribes | 10 |

## 5. Alternative metaheuristics

This paper utilizes 10 other metaheuristic algorithms as alternative approaches for comparative purposes. The GA, PSO, ACO, ICA, and CSS are selected as classical methods which have been utilized in most of the previous research, while the Butterfly Optimization Algorithm (BOA) [65], Harris Hawks Optimization (HHO) [66], Multi-Verse Optimizer (MVO) [67], Galactic Swarm Optimization (GSO) [68], and Krill Herd Algorithm (KHA) [69] are selected as some of the recently developed novel metaheuristic algorithms. Some of the approaches are classified as parameter-less optimization algorithms, meaning they do not have any internal parameters in their general formulation. At the same time, for some of them, some internal parameters need to be determined in the optimization process. In Table 2, a parameter summary is provided for these alternative approaches alongside the HS and the proposed Tribe-HS algorithms, while for all of them, the initial population size is utilized as 50.

## 6. Numerical results

The numerical outcomes of the weight optimization procedure for the 3-, 20- and 60-story steel structures are reported in this section. For each of the HS, Tribe-HS, and considered alternative methods, a total of 30 independent runs were undertaken. Figs. 13 to 15 show the convergence history for the best results of these approaches for the chosen 3-, 20-, and 60-story steel structures, respectively. It is worth noting that the Tribe-HS can get better outcomes than the standard HS with the minimum number of required structural analyses.

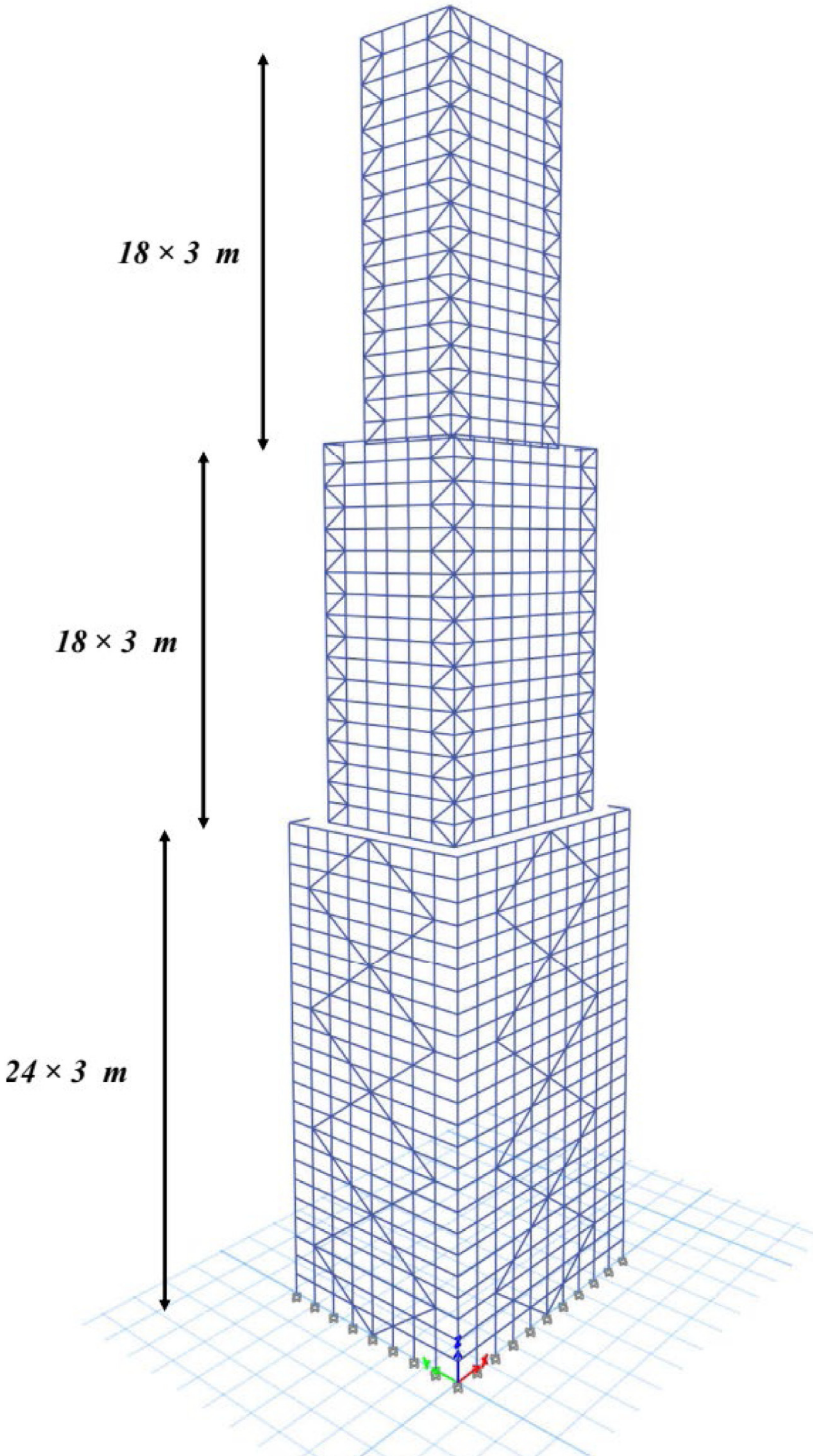


**Fig. 11.** Schematic view of the 60-story steel structure with 8272 members.

Table 3 presents the optimal design sections for the HS, Tribe-HS, and chosen alternative approaches when considering the three-story steel structure. It should be mentioned that for this reason, the best results from 30 independent runs in each metaheuristic method were reported.

The optimal design elements for the 20-story steel structures derived using the Tribe-HS and the conventional HS are provided in Table 4. The overall weight of the 20-story steel structure computed using the HS standard method is 3236.38 tons, but the Tribe-HS algorithm calculates this value as 2809.63 tons, which is less than the HS determined value. It could be observed that the overall weight of the structure acquired using Tribe-HS is less than the weight obtained using HS, demonstrating the suggested Tribe-HS method's capabilities for this purpose.

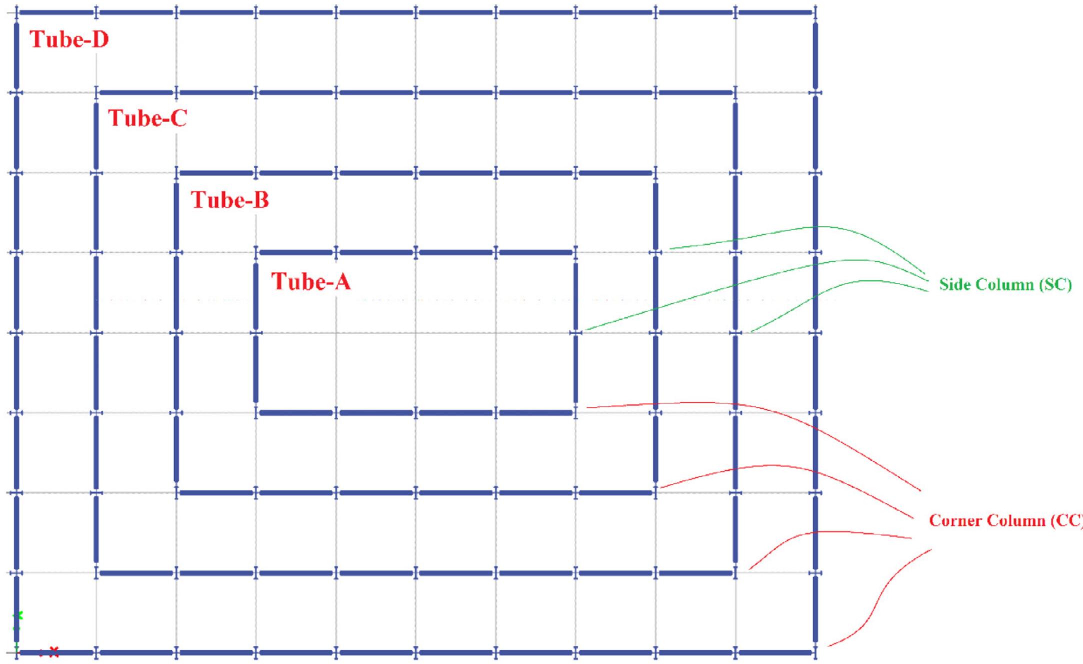


**Fig. 12.** Column grouping in plan levels of the 60-story steel structure with 8272 members.

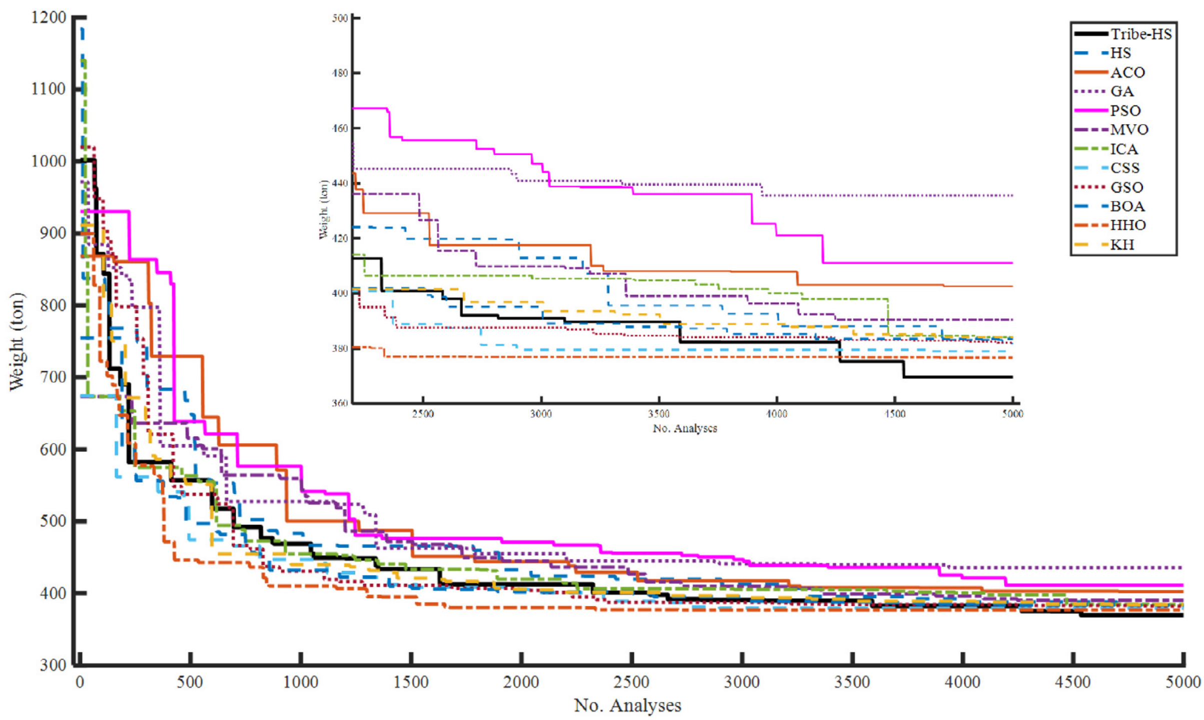


**Fig. 13.** Convergence history for the best results of different metaheuristics for 3-story structure.

Given that Kazemzadeh Azad, et al. [32] evaluated this design example using a variety of metaheuristic approaches, Table 5 compares the HS and the proposed Tribe-HS with various approaches. It should be highlighted that the Tribe-HS approach has the potential to provide superior outcomes than the other alternatives.

The optimal design sections for the 60-story steel structure derived using the Tribe-HS, and the conventional HS are provided in Table 6. The overall weight of the 60-story steel structure is determined using the HS standard method to be 6958.17 tons, whereas the Tribe-HS algorithm calculates it to be 6766.89 tons, which is less than the

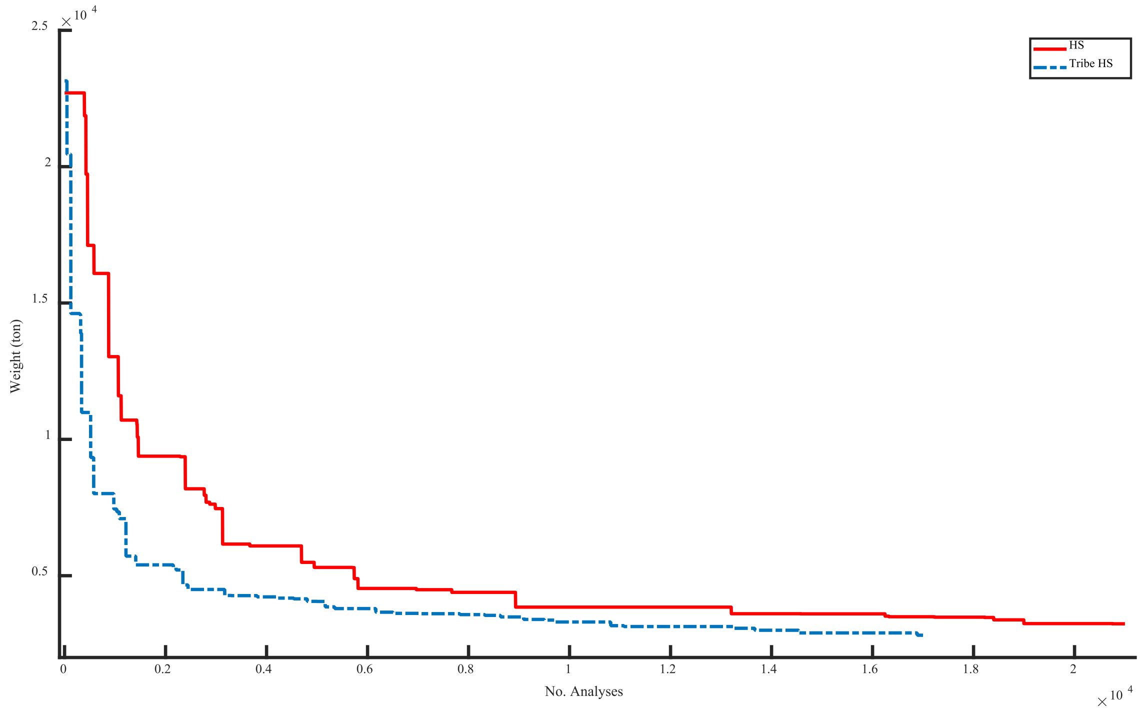


**Fig. 14.** Convergence history for the HS and Tribe-HS of the 20-story structure.

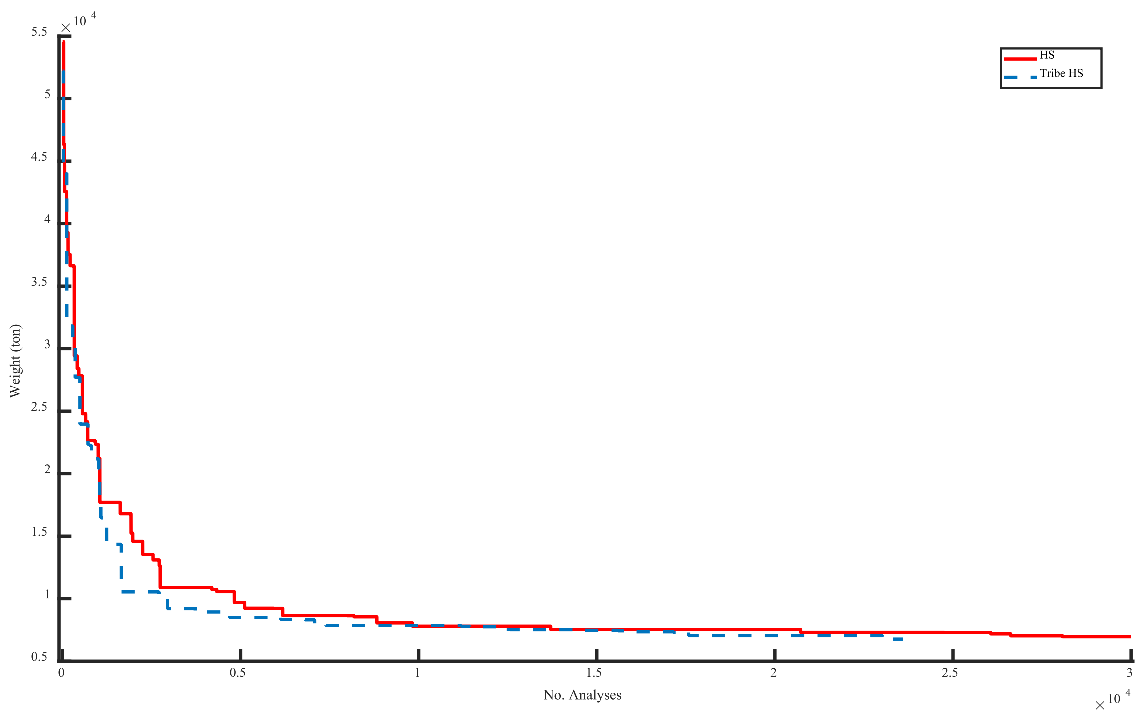


**Fig. 15.** Convergence history for the HS and Tribe-HS of the 60-story structure.

HS computed value. It could be mentioned that the overall weight of the structures acquired using Tribe-HS is less than that achieved using HS, demonstrating the suggested Tribe-HS method's capabilities. A comparative analysis is not appropriate since this instance is being described for the first time in this work.

The stress ratio of the structural elements for the 3, 20-and 60-story design examples are depicted in Figs. 16 and 18 respectively for the standard HS and the proposed Tribe-HS algorithms. The stress ratios of structural components in the Tribe-HS optimized structural systems are greater, particularly near the allowable value, demonstrat-

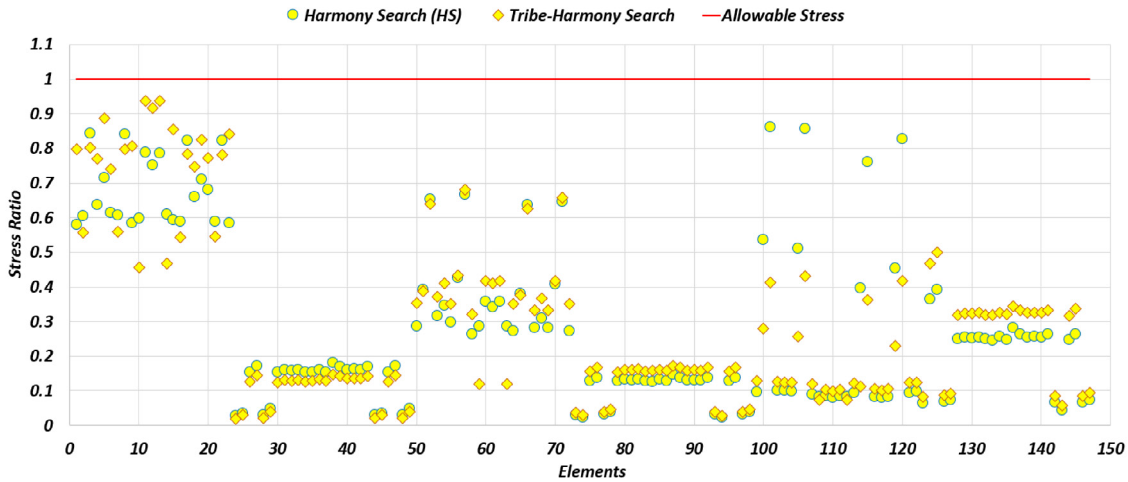


**Fig. 16.** Stress ratio of the structural elements for the 3-story design example.

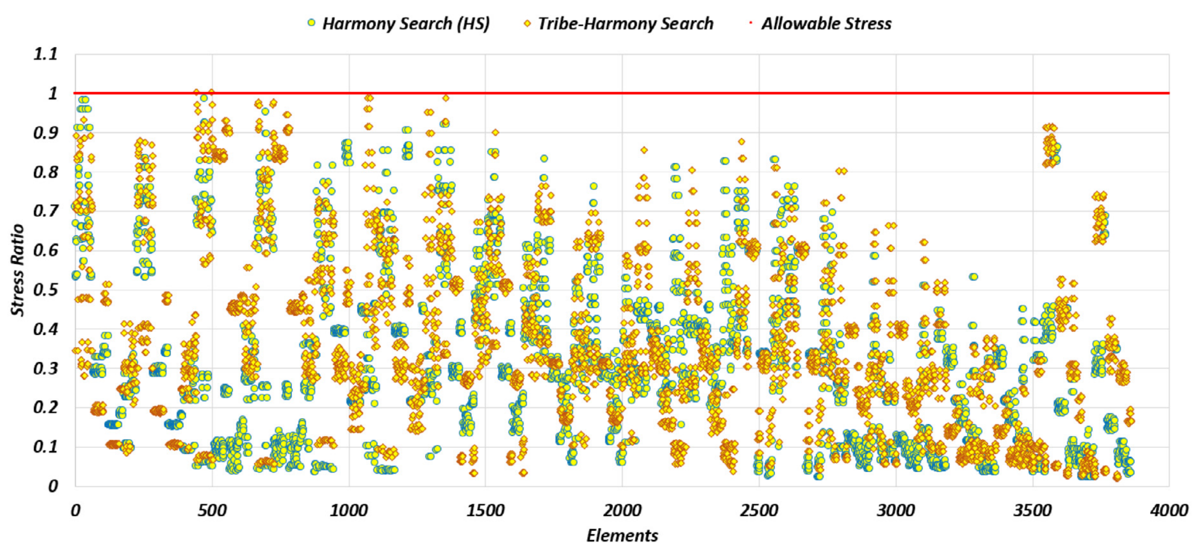


**Fig. 17.** The stress ratio of the structural elements for the 20-story design example.

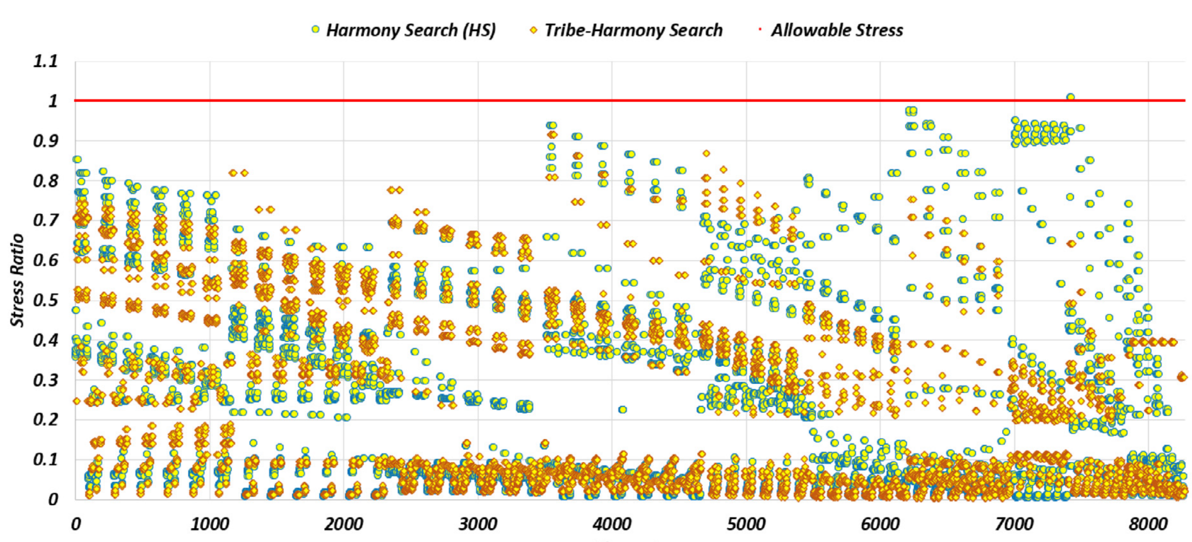


**Fig. 18.** The stress ratio of the structural elements for the 60-story design example.

**Table 3**
Optimum design sections of different metaheuristics for the 3-story steel structure.

| Groups | GA | PSO | ACO | ICA | MVO | GSO | BOA | KHA | HHO | HS | Tribe-HS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $CG_1$ | W8 × 28 | W18 × 60 | W14 × 38 | W16 × 36 | W24 × 55 | W24 × 68 | W12 × 35 | W10 × 33 | W18 × 55 | W14 × 43 | W16 × 40 |
| $CG_2$ | W30 × 90 | W16 × 67 | W12 × 45 | W14 × 48 | W18 × 60 | W21 × 62 | W16 × 57 | W21 × 62 | W18 × 55 | W21 × 68 | W18 × 60 |
| $CG_3$ | W24 × 76 | W27 × 84 | W24 × 68 | W18 × 50 | W27 × 94 | W27 × 84 | W40 × 149 | W21 × 73 | W18 × 55 | W24 × 68 | W30 × 90 |
| $CG_4$ | W33 × 130 | W30 × 90 | W27 × 84 | W30 × 99 | W24 × 94 | W24 × 84 | W24 × 68 | W24 × 76 | W30 × 90 | W21 × 73 | W21 × 62 |
| $B_1$ | W21 × 44 | W18 × 40 | W21 × 68 | W21 × 57 | W18 × 35 | W21 × 44 | W16 × 31 | W18 × 50 | W21 × 44 | W18 × 40 | W21 × 44 |
| $B_2$ | W8 × 18 | W18 × 35 | W18 × 40 | W16 × 40 | W18 × 40 | W16 × 26 | W24 × 55 | W18 × 46 | W16 × 36 | W21 × 44 | W18 × 40 |
| $B_3$ | W21 × 50 | W14 × 30 | W10 × 22 | W12 × 26 | W16 × 26 | W16 × 26 | W8 × 21 | W14 × 22 | W16 × 26 | W14 × 22 | W10 × 22 |
| $BR_1$ | W12 × 26 | W6 × 25 | W8 × 28 | W8 × 24 | W8 × 28 | W8 × 24 | W6 × 25 | W8 × 24 | W12 × 30 | W12 × 30 | W8 × 28 |
| $BR_2$ | W5 × 16 | W10 × 39 | W6 × 20 | W8 × 18 | W6 × 20 | W6 × 15 | W6 × 15 | W5 × 16 | W6 × 15 | W8 × 21 | W8 × 21 |
| $BR_3$ | W5 × 19 | W8 × 18 | W10 × 30 | W8 × 18 | W6 × 15 | W5 × 19 | W6 × 15 | W4 × 13 | W5 × 16 | W10 × 19 | W6 × 15 |
| Weight (ton) | 43.2073 | 41.1092 | 40.2583 | 38.3812 | 39.0215 | 38.2119 | 38.3569 | 38.1377 | 37.6269 | 38.1889 | 36.9721 |
| Maximum drift ratio | 1 | 0.9978 | 0.9599 | 1 | 1 | 0.9923 | 0.9915 | 1 | 1 | 0.9858 | 0.9965 |
| Maximum stress ratio | 1 | 0.9812 | 0.9505 | 0.9766 | 0.9683 | 0.9327 | 0.9971 | 1 | 0.9638 | 0.8613 | 0.9380 |

**CG:** Column Groups
**B:** Beam Group.
**BR:** Bracing Group.

**Table 4**
Optimum design sections for the 20-story steel structure with 3860 members.

| Stories | Groups | HS sections | Tribe-HS sections | Stories | Groups | HS sections | Tribe-HS sections |
|---|---|---|---|---|---|---|---|
| 1–2 | $CG_1$ | W12 × 50 | W30 × 99 | 11–12 | $CG_1$ | W24 × 68 | W33 × 291 |
| | $CG_2$ | W33 × 201 | W14 × 132 | | $CG_2$ | W21 × 132 | W27 × 94 |
| | $CG_3$ | W21 × 182 | W12 × 230 | | $CG_3$ | W18 × 311 | W30 × 124 |
| | $CG_4$ | W14 × 283 | W36 × 650 | | $CG_4$ | W27 × 194 | W36 × 300 |
| | $CG_5$ | W24 × 229 | W36 × 160 | | $CG_5$ | W27 × 235 | W18 × 76 |
| | IB | W14 × 82 | W10 × 100 | | IB | W21 × 83 | W24 × 84 |
| | OB | W14 × 74 | W14 × 61 | | OB | W8 × 67 | W12 × 72 |
| | BR | W30 × 173 | W18 × 119 | | BR | W14 × 26 | W14 × 109 |
| 3–4 | $CG_1$ | W40 × 249 | W24 × 207 | 13–14 | $CG_1$ | W44 × 262 | W10 × 77 |
| | $CG_2$ | W33 × 169 | W14 × 109 | | $CG_2$ | W21 × 132 | W33 × 130 |
| | $CG_3$ | W36 × 160 | W27 × 235 | | $CG_3$ | W12 × 87 | W18 × 143 |
| | $CG_4$ | W14 × 730 | W44 × 230 | | $CG_4$ | W27 × 161 | W18 × 106 |
| | $CG_5$ | W33 × 130 | W36 × 182 | | $CG_5$ | W18 × 76 | W14 × 68 |
| | IB | W33 × 152 | W12 × 50 | | IB | W8 × 58 | W12 × 53 |
| | OB | W24 × 103 | W14 × 48 | | OB | W24 × 207 | W36 × 135 |
| | BR | W18 × 86 | W12 × 65 | | BR | W8 × 48 | W14 × 53 |
| 5–6 | $CG_1$ | W14 × 176 | W12 × 72 | 15–16 | $CG_1$ | W27 × 161 | W44 × 335 |
| | $CG_2$ | W30 × 132 | W21 × 122 | | $CG_2$ | W18 × 258 | W36 × 260 |
| | $CG_3$ | W12 × 210 | W40 × 264 | | $CG_3$ | W36 × 182 | W40 × 174 |
| | $CG_4$ | W14 × 193 | W40 × 264 | | $CG_4$ | W44 × 262 | W27 × 84 |
| | $CG_5$ | W14 × 233 | W18 × 143 | | $CG_5$ | W18 × 97 | W30 × 90 |
| | IB | W10 × 68 | W27 × 84 | | IB | W21 × 122 | W14 × 68 |
| | OB | W18 × 55 | W21 × 73 | | OB | W21 × 147 | W24 × 103 |
| | BR | W36 × 182 | W8 × 40 | | BR | W8 × 40 | W10 × 45 |
| 7–8 | $CG_1$ | W14 × 68 | W40 × 183 | 17–18 | $CG_1$ | W24 × 62 | W30 × 477 |
| | $CG_2$ | W12 × 106 | W12 × 106 | | $CG_2$ | W27 × 258 | W12 × 79 |
| | $CG_3$ | W30 × 173 | W21 × 166 | | $CG_3$ | W40 × 321 | W16 × 67 |
| | $CG_4$ | W33 × 152 | W36 × 393 | | $CG_4$ | W24 × 94 | W40 × 174 |
| | $CG_5$ | W14 × 132 | W30 × 108 | | $CG_5$ | W18 × 175 | W10 × 88 |
| | IB | W18 × 76 | W12 × 58 | | IB | W21 × 93 | W14 × 99 |
| | OB | W14 × 68 | W21 × 166 | | OB | W12 × 87 | W30 × 116 |
| | BR | W24 × 103 | W24 × 94 | | BR | W8 × 40 | W14 × 68 |
| 9–10 | $CG_1$ | W18 × 60 | W16 × 77 | 19–20 | $CG_1$ | W18 × 158 | W24 × 250 |
| | $CG_2$ | W24 × 104 | W27 × 94 | | $CG_2$ | W27 × 161 | W10 × 112 |
| | $CG_3$ | W33 × 221 | W30 × 292 | | $CG_3$ | W16 × 50 | W18 × 86 |
| | $CG_4$ | W21 × 182 | W40 × 174 | | $CG_4$ | W40 × 174 | W8 × 67 |
| | $CG_5$ | W21 × 101 | W40 × 211 | | $CG_5$ | W12 × 152 | W24 × 104 |
| | IB | W27 × 84 | W10 × 77 | | IB | W16 × 67 | W18 × 55 |
| | OB | W14 × 109 | W27 × 102 | | OB | W16 × 57 | W12 × 190 |
| | BR | W27 × 94 | W10 × 49 | | BR | W10 × 49 | W10 × 17 |
| Total weight (ton) | | **3236.38** | **2809.63** | | | | |
| Maximum drift | | **0.8819** | **0.9091** | | | | |

**$CG_{1-5}$**: Column Groups 1 to 5 (Fig. 7).
**IB:** Inner Beam Group.
**OB:** Outer Beam Group.
**BR:** Bracing Group.

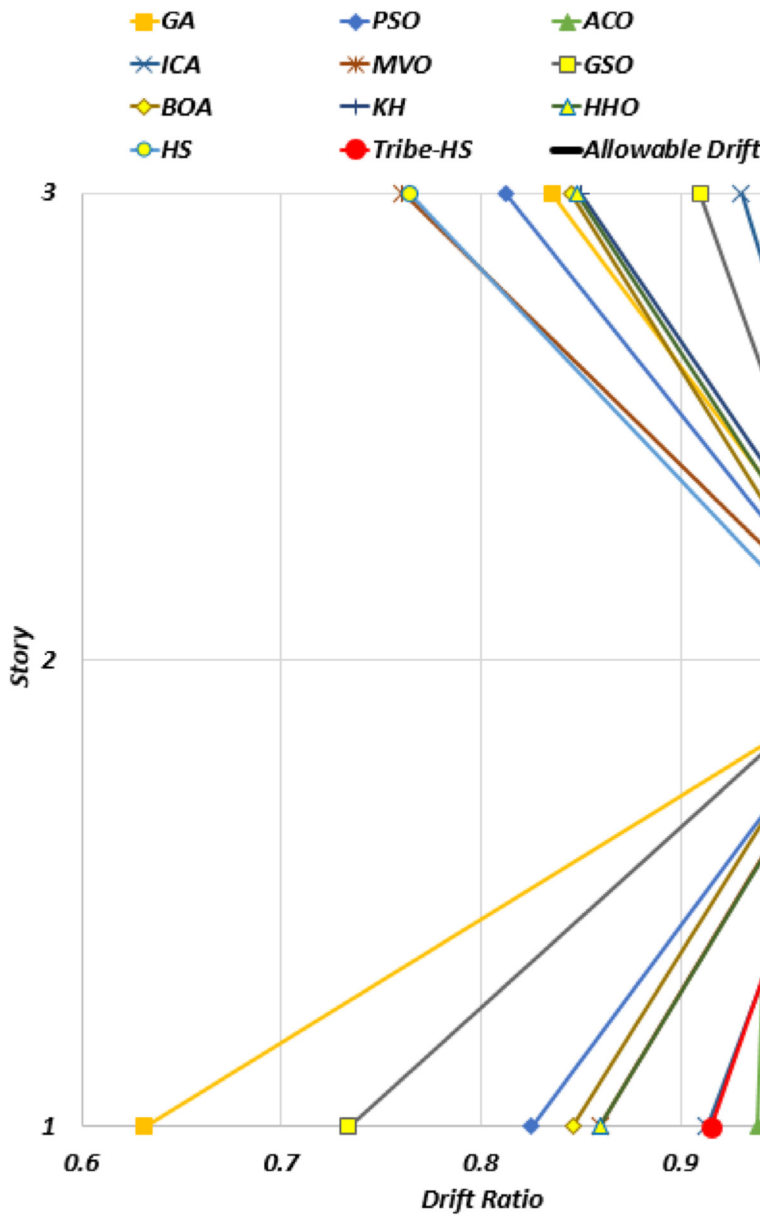


**Fig. 19.** Drift ratio of the structural elements for the 3-story design example.

**Table 5**
Comparative results for the 20-story steel structure with 3860 members.

| Stories | UBS [25] | HS | Tribe-HS |
|---|---|---|---|
| Total weight (ton) | 4117.43 | 3236.38 | 2809.63 |

**UBS:** Upper Bound Strategy.

ing that the Tribe-HS offered optimal design sections have the lowest feasible design cross-sections in terms of an affordable design approach (see Fig. 17).

The drift ratios of structural elements for the 3, 20, and 60 story design examples are shown in Figs. 19 to 21 for the standard HS and suggested Tribe-HS algorithms, respectively. For Tribe-HS-optimized structural systems, drift ratios of structural elements are greater, particularly near the allowable value, demonstrating that the Tribe-HS-provided optimal design sections have the smallest feasible design cross-sections in terms of an affordable design approach.

## 7. Conclusion

This study proposes an improved metaheuristic method named "Tribe-Harmony Search" for optimal steel structure design. This algorithm is a modified variant of the regular Harmony Search algorithm. The Harmony Search algorithm is one the wellknown metaheuristic algorithms utilizing the musical process of looking for the optimal state of harmony to produce an appropriate searching strategy. Due to the algorithm's many uses in a variety of optimization domains, there

**Table 6**
Optimum design sections for the 60-story steel structure with 8272 members.

| Stories | Groups | HS sections | Tribe-HS sections | Stories | Groups | HS sections | Tribe-HS sections |
|---|---|---|---|---|---|---|---|
| | CC-A | W27 × 94 | W24 × 279 | | CC-A | W36 × 280 | W30 × 235 |
| | SC-A | W21 × 111 | W21 × 57 | | SC-A | W14 × 257 | W40 × 199 |
| | CC-B | W27 × 84 | W21 × 93 | | CC-B | W27 × 84 | W16 × 67 |
| | SC-B | W24 × 55 | W21 × 62 | | SC-B | W16 × 50 | W21 × 50 |
| | CC-C | W21 × 147 | W14 × 550 | | CC-C | W14 × 145 | W36 × 393 |
| | SC-C | W40 × 277 | W24 × 492 | | SC-C | W36 × 328 | W36 × 280 |
| 1–6 | CC-D | W36 × 194 | W30 × 235 | 7–12 | CC-D | W18 × 311 | W14 × 211 |
| | SC-D | W40 × 249 | W36 × 245 | | SC-D | W33 × 354 | W14 × 283 |
| | BM-A | W30 × 148 | W18 × 192 | | BM-A | W24 × 492 | W30 × 148 |
| | BM-B | W40 × 199 | W18 × 258 | | BM-B | W33 × 141 | W10 × 100 |
| | BM-C | W27 × 129 | W24 × 408 | | BM-C | W18 × 55 | W16 × 45 |
| | BM-D | W40 × 211 | W27 × 114 | | BM-D | W33 × 130 | W12 × 96 |
| | BR-D | W18 × 65 | W24 × 68 | | BR-D | W24 × 103 | W24 × 94 |
| | CC-A | W21 × 132 | W24 × 131 | | CC-A | W24 × 94 | W21 × 132 |
| | SC-A | W24 × 103 | W36 × 135 | | SC-A | W40 × 199 | W24 × 84 |
| | CC-B | W14 × 159 | W27 × 258 | | CC-B | W21 × 101 | W30 × 124 |
| | SC-B | W33 × 318 | W40 × 277 | | SC-B | W27 × 178 | W12 × 279 |
| | CC-C | W24 × 162 | W24 × 176 | | CC-C | W30 × 261 | W18 × 97 |
| | SC-C | W36 × 260 | W27 × 217 | | SC-C | W36 × 245 | W40 × 297 |
| 13–18 | CC-D | W36 × 280 | W12 × 79 | 19–24 | CC-D | W30 × 99 | W27 × 129 |
| | SC-D | W30 × 326 | W18 × 211 | | SC-D | W40 × 174 | W21 × 147 |
| | BM-A | W14 × 43 | W40 × 167 | | BM-A | W12 × 152 | W21 × 101 |
| | BM-B | W36 × 280 | W14 × 193 | | BM-B | W14 × 132 | W40 × 199 |
| | BM-C | W36 × 170 | W21 × 122 | | BM-C | W16 × 77 | W18 × 158 |
| | BM-D | W21 × 44 | W30 × 90 | | BM-D | W18 × 35 | W27 × 84 |
| | BR-D | W27 × 102 | W27 × 94 | | BR-D | W24 × 55 | W27 × 146 |
| | CC-A | W40 × 167 | W33 × 130 | | CC-A | W36 × 182 | W27 × 258 |
| | SC-A | W12 × 120 | W24 × 117 | | SC-A | W14 × 132 | W44 × 230 |
| | CC-B | W27 × 235 | W21 × 147 | | CC-B | W14 × 132 | W12 × 336 |
| | SC-B | W30 × 124 | W18 × 86 | | SC-B | W24 × 104 | W30 × 124 |
| | CC-C | W40 × 277 | W30 × 292 | | CC-C | W40 × 174 | W30 × 211 |
| 25–30 | SC-C | W33 × 118 | W21 × 68 | 31–36 | SC-C | W14 × 120 | W40 × 211 |
| | BM-A | W24 × 146 | W12 × 136 | | BM-A | W21 × 166 | W36 × 359 |
| | BM-B | W40 × 235 | W14 × 74 | | BM-B | W14 × 90 | W12 × 30 |
| | BM-C | W21 × 68 | W40 × 167 | | BM-C | W24 × 76 | W33 × 263 |
| | BR-C | W36 × 182 | W27 × 161 | | BR-C | W40 × 264 | W24 × 103 |
| | CC-A | W30 × 132 | W33 × 169 | | CC-A | W36 × 245 | W14 × 159 |
| | SC-A | W10 × 112 | W40 × 199 | | SC-A | W36 × 160 | W21 × 166 |
| | CC-B | W40 × 431 | W30 × 108 | | CC-B | W44 × 290 | W14 × 159 |
| | SC-B | W14 × 398 | W40 × 215 | | SC-B | W14 × 370 | W12 × 210 |
| 37–42 | CC-C | W12 × 65 | W14 × 398 | 43–48 | BM-A | W24 × 207 | W27 × 146 |
| | SC-C | W27 × 178 | W30 × 99 | | BM-B | W14 × 53 | W40 × 183 |
| | BM-A | W27 × 194 | W40 × 174 | | BR-B | W21 × 182 | W40 × 149 |
| | BM-B | W12 × 96 | W18 × 50 | | – | | |
| | BM-C | W8 × 24 | W16 × 57 | | – | | |
| | BR-C | W33 × 318 | W14 × 61 | | – | | |
| | CC-A | W14 × 550 | W18 × 55 | | CC-A | W8 × 40 | W8 × 31 |
| | SC-A | W18 × 50 | W36 × 182 | | SC-A | W18 × 40 | W12 × 252 |
| | CC-B | W36 × 150 | W18 × 40 | | CC-B | W40 × 149 | W27 × 161 |
| 49–54 | SC-B | W24 × 76 | W24 × 117 | 55–60 | SC-B | W12 × 35 | W18 × 119 |
| | BM-A | W18 × 97 | W36 × 170 | | BM-A | W16 × 67 | W12 × 50 |
| | BM-B | W16 × 67 | W12 × 152 | | BM-B | W21 × 68 | W12 × 96 |
| | BR-B | W36 × 170 | W8 × 35 | | BR-B | W33 × 221 | W30 × 173 |
| Total Weight (ton) | | **6958.17** | **6766.89** | | | | |
| Maximum Drift | | **0.9985** | **0.9722** | | | | |

**CC-A, CC-B, CC-C, CC-D:** Corner Column Groups for Tubes A to D (Fig. 9).
**SC-A, SC-B, SC-C, SC-D:** Side Column Groups for Tubes A to D (Fig. 9).
**BM-A, BM-B, BM-C, BM-D:** Beam Member Groups for Tubes A to D.
**BR-B, BR-C, BR-D:** Bracing Member Groups for Tubes B to D.

has been an increased interest in improving the algorithm's overall performance. The traditional algorithm's searching phase is broken into three distinct phases in the Tribe-HS. These stages, called tribes, lead the algorithm to prioritize global search in the early iterations and local search in the later iterations. These adjustments improve the conventional algorithm's exploration and exploitation rates. 3 different building structures with 3, 20, and 60 stories with 135, 3860, and 8272 structural members are deemed as design examples to assess the ability of the suggested methodology in dealing with complex optimization problems. The suggested method's overall performance is compared to that of the standard Harmony Search algorithm and many meta-heuristics. The acquired findings demonstrated that the recommended technique is capable of producing superior outcomes than the other metaheuristics for the investigated design examples. The total weight of the 20-story steel structure is obtained as 3236.38 tons using HS and 2809.63 tons using Tribe-HS, while the reduction rate is about 13%.

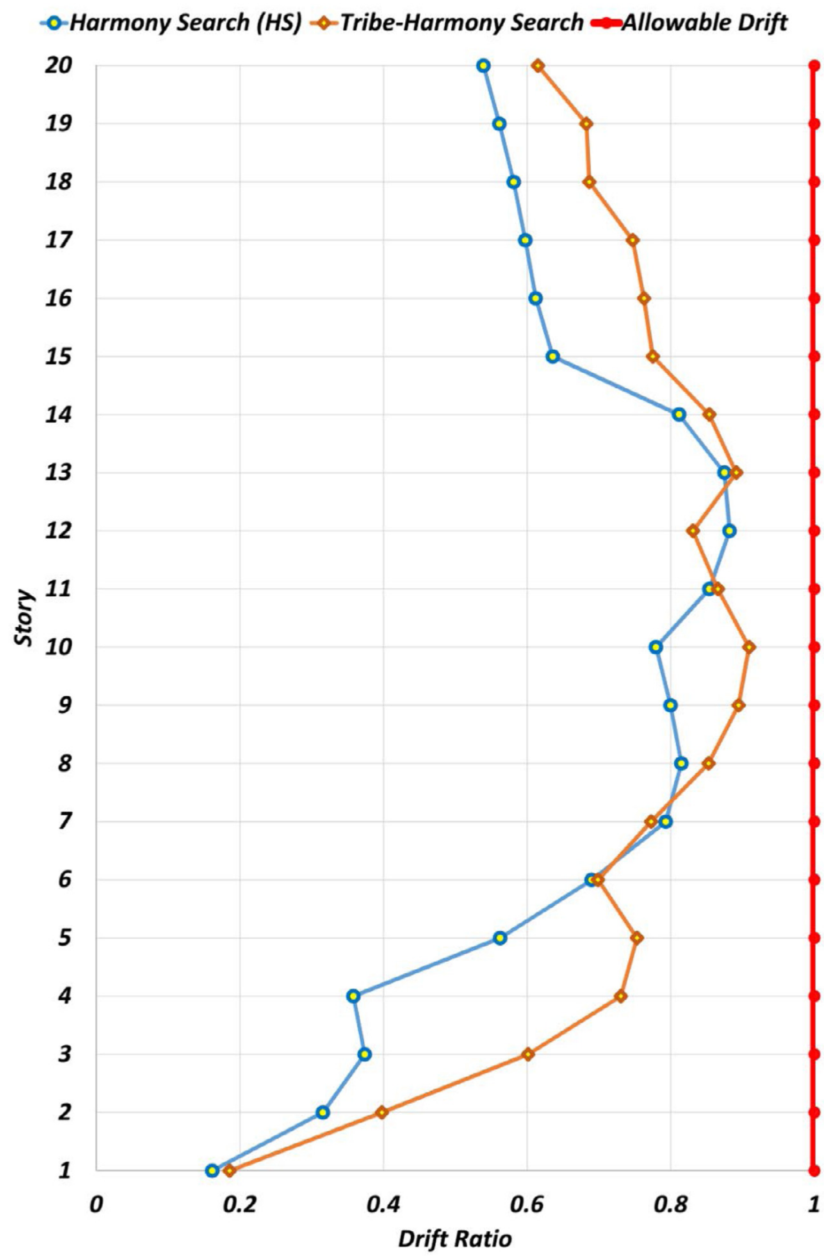


**Fig. 20.** The drift ratio of the structural elements for the 20-story design example.

The overall weight of the 60-story steel structure is 6958.17 tons when using HS and 6766.89 tons when employing Tribe-HS, with a 3 percent decrease rate. The stress and drift ratios of the structural elements are higher in the Tribe-HS optimized structural systems, particularly near the allowable value, demonstrating that the Tribe-HS provided optimum design sections have the lowest possible design cross-sections in terms of an economical design process.

## Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.dajour.2022.100067. This research is supported by a research grant from the University of Tabriz (Number: 1615)

## Acknowledgments

This research is supported by a research grant from the University of Tabriz, Iran (Number: 1615).

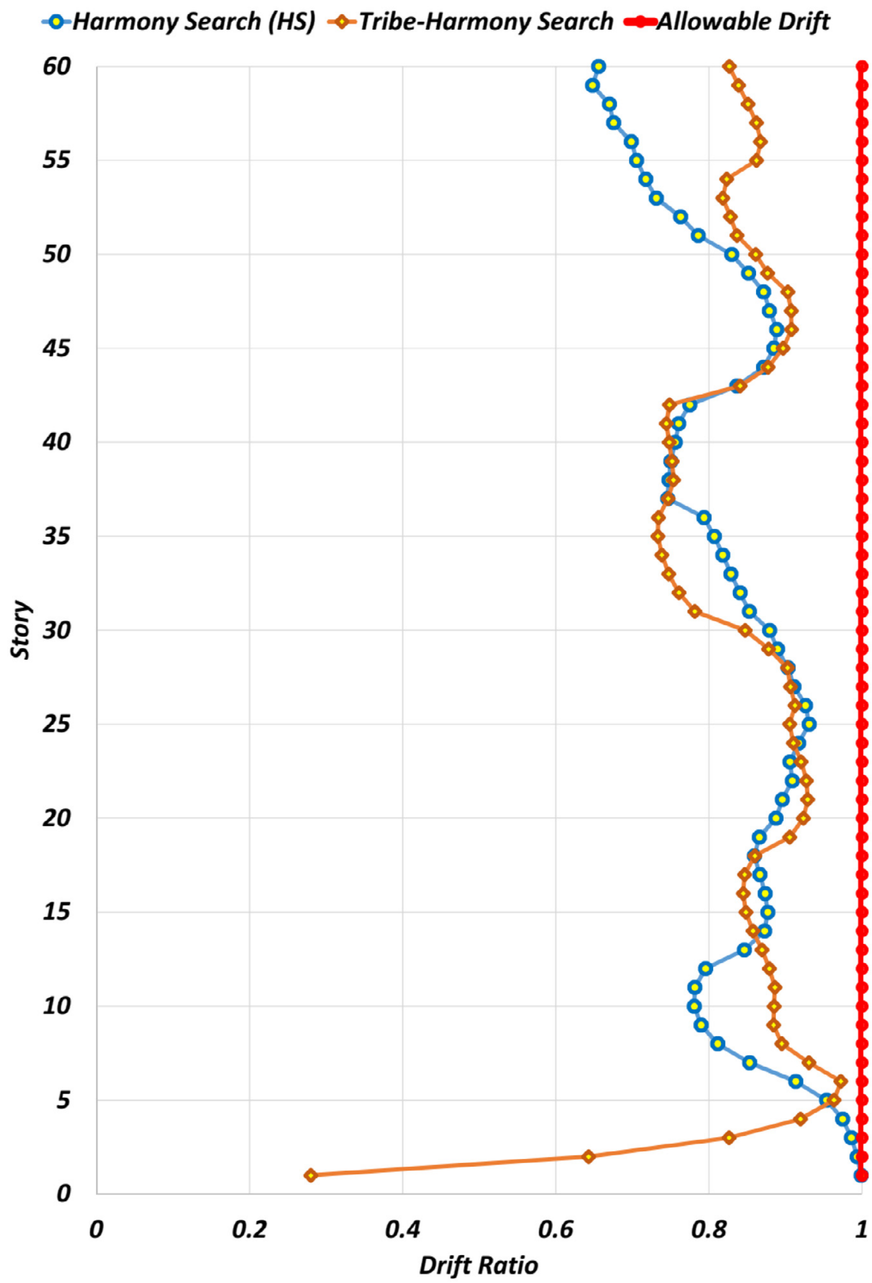


**Fig. 21.** The drift ratio of the structural elements for the 60-story design example.

## References


[1] J.H. Holland, Genetic algorithms and adaptation, in: O.G. Selfridge, E.L. Rissland, M.A. Arbib (Eds.), Adaptive Control of Ill-Defined Systems, Springer US, Boston, MA, 1984, pp. 317–333.

[2] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[3] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1) (1996) 29–41, http://dx.doi.org/10.1109/3477.484436.

[4] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, in: MHS'95, 1995, pp. 39–43.

[5] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289, http://dx.doi.org/10.1007/s00707-009-0270-4.

[6] S. Talatahari, M. Azizi, M. Toloo, M. Baghalzadeh Shishehgarkhaneh, Optimization of large-scale frame structures using fuzzy adaptive quantum inspired charged system search, Int. J. Steel Struct. (2022) http://dx.doi.org/10.1007/s13296-022-00598-y.

[7] S. Talatahari, M. Azizi, M. Toloo, Fuzzy adaptive charged system search for global optimization, Appl. Soft Comput. 109 (2021) 107518.

[8] S. Talatahari, M. Azizi, A.H. Gandomi, Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems, Processes 9 (5) (2021) 859.

[9] M. Azizi, M.B. Shishehgarkhaneh, M. Basiri, Optimum design of truss structures by material generation algorithm with discrete variables, Decis. Anal. J. (2022) 100043, http://dx.doi.org/10.1016/j.dajour.2022.100043.

[10] S. Talatahari, M. Azizi, Chaos game optimization: a novel metaheuristic algorithm, Artif. Intell. Rev. 54 (2) (2021) 917–1004, http://dx.doi.org/10.1007/s10462-020-09867-w.

[11] M. Azizi, U. Aickelin, H.A. Khorshidi, M.B. Shishehgarkhaneh, Shape and size optimization of truss structures by chaos game optimization considering frequency constraints, J. Adv. Res. (2022) http://dx.doi.org/10.1016/j.jare.2022.01.002.

[12] G.-G. Wang, A.H. Gandomi, X.-S. Yang, A.H. Alavi, A novel improved accelerated particle swarm optimization algorithm for global numerical optimization, Eng. Comput. (2014).

[13] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. (2010).

[14] M. Azizi, R.G. Ejlali, S.A. Mousavi Ghasemi, S. Talatahari, Upgraded whale optimization algorithm for fuzzy logic based vibration control of nonlinear steel

structure, Eng. Struct. 192 (2019) 53–70, http://dx.doi.org/10.1016/j.engstruct.2019.05.007.

[15] F. Fasahat, P. Payvandy, A novel hybrid genetic and imperialist competitive algorithm for structure extraction of woven fabric images, J. Textile Inst. 108 (6) (2017) 893–905.

[16] M. Azizi, A. Mousavi, R. Ejlali, S. Talatahari, Optimum design of fuzzy controller using hybrid ant lion optimizer and Jaya algorithm, Artif. Intell. Rev. 53 (2020) 1–32, http://dx.doi.org/10.1007/s10462-019-09713-8.

[17] M. Azizi, Atomic orbital search: A novel metaheuristic algorithm, Appl. Math. Model. 93 (2021) 657–683, http://dx.doi.org/10.1016/j.apm.2020.12.021.

[18] S. Talatahari, M. Azizi, Optimization of constrained mathematical and engineering design problems using chaos game optimization, Comput. Ind. Eng. 145 (2020) 106560, http://dx.doi.org/10.1016/j.cie.2020.106560.

[19] S. Talatahari, M. Azizi, Optimal design of real-size building structures using quantum-behaved developed swarm optimizer, Struct. Des. Tall Special Build. 29 (11) (2020) e1747.

[20] S. Talatahari, M. Azizi, Optimum design of building structures using tribe-interior search algorithm, Structures 28 (2020) 1616–1633, http://dx.doi.org/10.1016/j.istruc.2020.09.075.

[21] S. Talatahari, M. Azizi, Tribe-charged system search for global optimization, Appl. Math. Model. 93 (2021) 115–133, http://dx.doi.org/10.1016/j.apm.2020.12.007.

[22] S. Talatahari, M. Azizi, An extensive review of charged system search algorithm for engineering optimization applications, in: S. Carbas, A. Toktas, D. Ustun (Eds.), Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications, Springer Singapore, Singapore, 2021, pp. 309–334.

[23] M. Azizi, Designing fuzzy controllers for frame structures based on ground motion prediction using grasshopper optimization algorithm: A case study of Tabriz, Iran, in: S. Carbas, A. Toktas, D. Ustun (Eds.), Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications, Springer Singapore, Singapore, 2021, pp. 153–180.

[24] M. Azizi, U. Aickelin, H.A. Khorshidi, M. Baghalzadeh Shishehgarkhaneh, Shape and size optimization of truss structures by chaos game optimization considering frequency constraints, J. Adv. Res. (2022) http://dx.doi.org/10.1016/j.jare.2022.01.002.

[25] M. Farshchin, M. Maniat, C.V. Camp, S. Pezeshk, School based optimization algorithm for design of steel frames, Eng. Struct. 171 (2018) 326–335.

[26] N. Khodadadi, M. Azizi, S. Talatahari, P. Sareh, Multi-objective crystal structure algorithm (MOCryStAl): Introduction and performance evaluation, IEEE Access 9 (2021) 117795-117812, http://dx.doi.org/10.1109/ACCESS.2021.3106487.

[27] A. Kaveh, A. BolandGerami, Optimal design of large-scale space steel frames using cascade enhanced colliding body optimization, Struct. Multidiscip. Optim. 55 (1) (2017) 237–256.

[28] S. Talatahari, A.H. Gandomi, X.-S. Yang, S. Deb, Optimum design of frame structures using the eagle strategy with differential evolution, Eng. Struct. 91 (2015) 16–25, http://dx.doi.org/10.1016/j.engstruct.2015.02.026.

[29] M.R. Maheri, H. Shokrian, M. Narimani, An enhanced honey bee mating optimization algorithm for design of side sway steel frames, Adv. Eng. Softw. 109 (2017) 62–72.

[30] A. Kaveh, M.I. Ghazaan, Enhanced whale optimization algorithm for sizing optimization of skeletal structures, Mech. Based Des. Struct. Mach. 45 (3) (2017) 345–362.

[31] S. Kazemzadeh Azad, O. Hasançebi, S. Kazemzadeh Azad, Upper bound strategy for metaheuristic based design optimization of steel frames, Adv. Eng. Softw. 57 (2013) 19–32.

[32] S. Kazemzadeh Azad, O. Hasançebi, S. Kazemzadeh Azad, Computationally efficient optimum design of large scale steel frames, Int. J. Optim. Civil Eng. 4 (2014).

[33] O. Hasançebi, Cost efficiency analyses of steel frameworks for economical design of multi-storey buildings, J. Construct. Steel Res. 128 (2017) 380–396.

[34] C. Tort, S. Şahin, O. Hasançebi, Optimum design of steel lattice transmission line towers using simulated annealing and PLS-TOWER, Comput. Struct. 179 (2017) 75–94.

[35] T. Kundu, H. Garg, A hybrid ITLHHO algorithm for numerical and engineering optimization problems, Int. J. Intell. Syst. http://dx.doi.org/10.1002/int.22707, n/a(n/a).

[36] A. Kaveh, Y. Vazirinia, An upgraded Sine cosine algorithm for tower crane selection and layout problem, Periodica Polytechn. Civil Eng. 64 (2) (2020) 325–343, http://dx.doi.org/10.3311/PPci.15363.

[37] I. Brajević, M. Tuba, An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems, J. Intell. Manuf. 24 (4) (2013) 729–740, http://dx.doi.org/10.1007/s10845-011-0621-6.

[38] I. Brajević, J. Ignjatović, An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems, J. Intell. Manuf. 30 (6) (2019) 2545–2574, http://dx.doi.org/10.1007/s10845-018-1419-6.

[39] V.K. Pathak, A.K. Srivastava, A novel upgraded bat algorithm based on cuckoo search and sugeno inertia weight for large scale and constrained engineering design optimization problems, Eng. Comput. (2020) http://dx.doi.org/10.1007/s00366-020-01127-3.

[40] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: Harmony search, Simulation 76 (2) (2001) 60–68, http://dx.doi.org/10.1177/003754970107600201.

[41] T. Zhang, Z.W. Geem, Review of harmony search with respect to algorithm structure, Swarm Evol. Comput. 48 (2019) 31–43, http://dx.doi.org/10.1016/j.swevo.2019.03.012.

[42] H. Rezaie, M.H. Kazemi-Rahbar, B. Vahidi, H. Rastegar, Solution of combined economic and emission dispatch problem using a novel chaotic improved harmony search algorithm, J. Comput. Des. Eng. 6 (3) (2019) 447–467.

[43] G. Li, X. Li, L. Gao, B. Zeng, Tasks assigning and sequencing of multiple AGVs based on an improved harmony search algorithm, J. Ambient Intell. Humaniz. Comput. 10 (11) (2019) 4533–4546.

[44] P. Monica, M. Kowsalya, Improved harmony search based optimization of droop control parameters for load sharing in DC microgrids, in: TENCON 2019-2019 IEEE Region 10 Conference (TENCON), IEEE, 2019, pp. 2199–2203.

[45] B.F. Zohra, A.K. Lahouari, R. Mostefa, An improved harmony search algorithm for solved the combined heat and power economic dispatch, Int. J. Electr. Eng. Inform. 11 (2) (2019) 440–450.

[46] W. Wu, H. Ouyang, A.W. Mohamed, C. Zhang, S. Li, Enhanced harmony search algorithm with circular region perturbation for global optimization problems, Appl. Intell. 50 (3) (2020) 951–975.

[47] A. Askarzadeh, M. Montazeri, L.D.S. Coelho, A modified harmony search algorithm applied to capacitor placement of radial distribution networks considering voltage stability index, Int. J. Bio-Inspired Comput. 13 (3) (2019) 189–198.

[48] K. Ji, W. Wang, X. Wang, Z. Wei, J. Yang, H. Qin, Optimal installation of distributed generators based on an enhanced harmony search algorithm, J. Phys. Conf. Ser. (2019) 022006, IOP Publishing.

[49] B. Keshtegar, P. Hao, Y. Wang, Y. Li, Optimum design of aircraft panels based on adaptive dynamic harmony search, Thin-Walled Struct. 118 (2017) 37–45, http://dx.doi.org/10.1016/j.tws.2017.05.004.

[50] H. Ouyang, W. Wu, C. Zhang, S. Li, D. Zou, G. Liu, Improved harmony search with general iteration models for engineering design optimization problems, Soft Comput. 23 (20) (2019) 10225–10260.

[51] Y.Y. Moon, Z.W. Geem, G.-T. Han, Vanishing point detection for self-driving car using harmony search algorithm, Swarm Evol. Comput. 41 (2018) 111–119, http://dx.doi.org/10.1016/j.swevo.2018.02.007.

[52] J. Yi, X. Li, C.-H. Chu, L. Gao, Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization, J. Intell. Manuf. 30 (1) (2019) 405–428, http://dx.doi.org/10.1007/s10845-016-1255-5.

[53] B. Keshtegar, T. Ozbakkaloglu, A. Gholampour, Modeling the behavior of FRP-confined concrete using dynamic harmony search algorithm, Eng. Comput. 33 (3) (2017) 415–430, http://dx.doi.org/10.1007/s00366-016-0481-y.

[54] M. Hasanipanah, B. Keshtegar, D.-K. Thai, N.-T. Troung, An ANN-adaptive dynamical harmony search algorithm to approximate the flyrock resulting from blasting, Eng. Comput. 38 (2) (2022) 1257–1269, http://dx.doi.org/10.1007/s00366-020-01105-9.

[55] J. Yi, L. Gao, X. Li, C.A. Shoemaker, C. Lu, An on-line variable-fidelity surrogate-assisted harmony search algorithm with multi-level screening strategy for expensive engineering design optimization, Knowl.-Based Syst. 170 (2019) 1–19.

[56] R. Sheikholeslami, A.C. Zecchin, F. Zheng, S. Talatahari, A hybrid cuckoo–harmony search algorithm for optimal design of water distribution systems, J. Hydroinform. 18 (3) (2016) 544–563.

[57] B. Keshtegar, P. Hao, Y. Wang, Q. Hu, An adaptive response surface method and Gaussian global-best harmony search algorithm for optimization of aircraft stiffened panels, Appl. Soft Comput. 66 (2018) 196–207, http://dx.doi.org/10.1016/j.asoc.2018.02.020.

[58] H. Ouyang, G. Ma, G. Liu, Z. Li, X. Zhong, Hybrid teaching-learning based optimization with harmony search for engineering optimization problems, in: 2017 36th Chinese Control Conference (CCC), IEEE, 2017, pp. 2714–2717.

[59] B. Keshtegar, S. Etedali, Nonlinear mathematical modeling and optimum design of tuned mass dampers using adaptive dynamic harmony search algorithm, Struct. Control Health Monit. 25 (7) (2018) e2163, http://dx.doi.org/10.1002/stc.2163.

[60] S. Gholizadeh, A. Barzegar, Shape optimization of structures for frequency constraints by sequential harmony search algorithm, Eng. Optim. 45 (6) (2013) 627–646.

[61] M. Jaberipour, E. Khorram, A new harmony search algorithm for solving mixed–discrete engineering optimization problems, Eng. Optim. 43 (5) (2011) 507–523.

[62] A. ANSI, ANSI/AISC 360-10, in: Specification for Structural Steel Buildings, American Institute of Steel Construction, Inc., Chicago, IL, 2010.

[63] K.S. Lee, Z.W. Geem, A new structural optimization method based on the harmony search algorithm, Comput. Struct. 82 (9–10) (2004) 781–798.

[64] ASCE, Minimum Design Loads for Buildings and Other Structures, American Society of Civil Engineers, 2013.

[65] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734.

[66] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872, http://dx.doi.org/10.1016/j.future.2019.02.028.

[67] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513, http://dx.doi.org/10.1007/s00521-015-1870-7.

[68] V. Muthiah-Nakarajan, M.M. Noel, Galactic swarm optimization: A new global optimization metaheuristic inspired by galactic motion, Appl. Soft Comput. 38 (2016) 771–787.

[69] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.