Journal of Civil Aspects and Structural Engineering

www.case.reapress.com

J. Civ. Asp. Struct. Eng. Vol. 3, No. 2 (2026) 143–156.

Paper Type: Original Article

A Hybrid Imperialist Competitive Algorithm and Harmony Search for Optimal Design of 2D Steel Frames

Masoud Saadatmand[1,*], **Taha Bakhshpoori**[2]

[1] Member of the Engineering Organization of Mazandaran Province, Mazandaran, Iran; bluetooth330@yahoo.com.
[2] Department of Engineering and Technology, East Guilan Branch, University of Guilan, Iran; Bakhshpoori.Ta@gmail.com.

Citation:

Received: 15 May 2025 *Revised: 26 July 2025* *Accepted: 10 October 2025*	Saadatmand, M., & Bakhshpoori, T. (2026). A hybrid imperialist competitive algorithm and harmony search for optimal design of 2D steel frames. *Journal of Civil Aspects and Structural Engineering, 3*(2), 143–156.

Abstract

Structural optimization remains a critical challenge in civil engineering, particularly for steel frame design under discrete variable constraints. This paper presents a novel hybrid metaheuristic algorithm combining the Imperialist Competitive Algorithm (ICA) and Harmony Search (HS) to optimize the weight of two-dimensional steel frames. The proposed ICA-HS hybrid leverages the social-political assimilation mechanism of ICA and the memory-based improvisation of HS to enhance convergence speed and solution accuracy. The algorithm is evaluated on two benchmark problems: A 15-story, three-bay steel frame and a 24-story, three-bay steel frame, both designed according to AISC-LRFD specifications using discrete W-section profiles. Comparative results against standard ICA, CSS, CBO, ECBO, HS, and HBB-BC algorithms demonstrate that ICA-HS achieves competitive optimal weights (88,246 lb for the 15-story frame and 202,517 lb for the 24-story frame) with superior convergence stability. Statistical analysis over 20 independent runs confirms the robustness of the proposed hybrid. The hybrid strategy effectively balances global exploration and local exploitation, overcoming the premature convergence limitations of standalone ICA.

Keywords: Steel frame optimization, Imperialist competitive algorithm, Harmony search, Metaheuristic, Discrete structural optimization, AISC-LRFD.

1 | Introduction

Optimization has become one of the most dynamic research areas in structural engineering, driven by rapid advances in computational hardware, user-friendly software platforms (e.g., MATLAB optimization toolbox), and the increasing demand for cost-effective, material-efficient designs. Structural optimization aims to select the "best" design—typically the lightest weight or lowest cost—from a set of feasible candidates while satisfying all design constraints (stress, displacement, stability, and code-specific requirements) [1].

Steel frames, widely used in mid- to high-rise buildings, present a particularly challenging optimization problem due to:

Corresponding Author: bluetooth330@yahoo.com

https://doi.org/10.48314/jcase.v3i2.83

I. Mixed continuous-discrete variable nature (W-section selection).

II. Large numbers of design variables (member groups).

III. Nonlinear constraints from interaction equations (axial + bending).

IV. Code-specific slenderness and drift limits.

Metaheuristic algorithms have emerged as powerful tools for such complex, non-convex, discrete optimization problems. Unlike gradient-based methods, metaheuristics do not require derivatives and can escape local optima through stochastic search mechanisms. Among the numerous metaheuristics developed over the past three decades—including Genetic Algorithms (GA) [2], [3] Particle Swarm Optimization (PSO) [4], Ant Colony Optimization (ACO) [5], and more recent algorithms like Colliding Bodies Optimization (CBO) [6] and Enhanced Charged System Search (ECSS) [7]—the Imperialist Competitive Algorithm (ICA) [8] and Harmony Search (HS) [9] have shown particular promise for structural applications.

ICA, inspired by socio-political imperialistic competition, has demonstrated competitive accuracy and robustness for steel frame optimization [10]. However, its primary drawback is relatively high computational cost and occasional premature convergence to local optima. HS, inspired by musical improvisation, offers fast convergence and efficient memory utilization but may struggle with local exploitation in discrete spaces [11].

This paper addresses these limitations by proposing a hybrid ICA-HS algorithm that integrates HS's memory-based search mechanism into the ICA framework. The hybrid aims to:

I. Accelerate convergence while maintaining solution accuracy.

II. Prevent premature entrapment in local optima.

III. Achieve lighter steel frame designs compared to standalone algorithms.

The remainder of this paper is organized as follows. Section 2 formulates the steel frame optimization problem. Section 3 describes the ICA, HS, and the proposed hybrid. Section 4 presents numerical results for 15- and 24-story frames. Section 5 discusses findings, and Section 6 concludes with recommendations for future research.

2 | Problem Formulation

2.1 | General Optimization Model

A constrained structural optimization problem can be expressed as

$$\min\ f(x)\ \text{subject to}\ g_j(x) \le 0, j = 1,2,\dots,m,$$
$$x_i \in S_i, i = 1,2,\dots,n,$$

where:

I. $f(x)$ = objective function (structural weight).

II. $x = [x_1, x_2, \dots, x_n]^T$ = vector of design variables (cross-sectional areas).

III. $g_j(x)$ = constraint functions (stress, drift, slenderness, interaction).

IV. S_i = allowable set of discrete sections (W-shapes).

2.2 | Objective Function

The objective is to minimize the total weight of the steel frame:

$$f(x) = \sum_{i=1}^{n} \rho \cdot A_i \cdot L_i,$$

where:

I. ρ = steel density (0.2836 lb/in³ for A36 steel).

II. A_i = cross-sectional area of member i.

III. L_i = length of member i.

IV. n = total number of members.

2.3 | Design Variables – Discrete

All members are selected from the AISC W-section database (267 sections for beams, 37 W14 sections for columns in the 24-story example). Thus, design variables are discrete:

$$x_i \in \{W_1, W_2, \dots, W_p\},$$

where p is the number of available W-sections.

2.4 | Constraint Handling – Penalty Function

A penalty function method is employed to handle violated constraints:

$$F(x) = f(x) \times \left(1+\varepsilon \sum_{j=1}^{m} \left[\max\left(0, g_j(x)\right)\right]^2\right),$$

where:

I. $F(x)$ = penalized objective function.

II. ε = penalty coefficient (typically 10–100).

III. $g_j(x) \leq 0$ represents feasible constraints.

2.5 | Design Constraints (Based on AISC-LRFD 1999)

2.5.1 | Stress constraints for truss elements (if applicable)

For tension members:

$$f_t \leq 0.9F_y,$$

for compression members:

$$f_a \leq F_{cr},$$

where:

I. f_t = axial tensile stress.

II. f_a = axial compressive stress.

III. F_y = yield stress.

IV. F_{cr} = critical buckling stress per AISC.

2.5.2 | Drift constraints – inter-story drift

$$\frac{\Delta_i}{h_i} \leq 0.0025 \text{ (or code-specified limit)},$$

where:

I. Δ_i = lateral displacement of story i.

II. h_i = height of story i.

2.5.3 | Beam-column interaction (axial + bending)

For $\frac{P_u}{\phi_c P_n} \geq 0.2$:

$$\frac{P_u}{\phi_c P_n} + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \leq 1.0,$$

For $\frac{P_u}{\phi_c P_n} < 0.2$:

$$\frac{P_u}{2\phi_c P_n} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \leq 1.0,$$

where:

I. Pu = required axial strength

II. P_n = nominal axial strength.

III. M_{ux}, M_{uy} = required flexural strengths about x and y axes.

IV. M_{nx}, M_{ny} = nominal flexural strengths.

V. $\phi_c = 0.85$ (compression), $\phi_b = 0.90$ (bending).

2.5.4 | Effective length factor (dumonteil formula)

$$G_A = \frac{\sum(EI/L)_c}{\sum(EI/L)_b}, G_B = \frac{\sum(EI/L)_c}{\sum(EI/L)_b},$$

then K is obtained from the alignment chart or:

$$K = \sqrt{\frac{\pi^2 EI}{P_{cr} L^2}},$$

for practical use, the following approximation for sway frames:

$$K = \frac{3G_A G_B + 1.4(G_A + G_B) + 0.64}{3G_A G_B + 2.0(G_A + G_B) + 1.28}.$$

2.5.5 | Slenderness limits

for tension members:

$$\frac{KL}{r} \leq 300,$$

for compression members:

$$\frac{KL}{r} \leq 200,$$

where:

I. K = effective length factor.

II. L = unbraced length.

III. r = radius of gyration.

3 | Hybrid ICA-HS Algorithm

3.1 | Overview of Imperialist Competitive Algorithm

ICA [6] is a population-based metaheuristic inspired by imperialistic competition. The algorithm steps (see flowchart Fig. 3-8 in original thesis):

Step 1. Initialization: Generate $N_{country}$ random countries (each country = a design vector).

Select N_{imp} strongest as imperialists; assign remaining colonies to imperialists proportionally to imperialist power.

Step 2. Assimilation (absorption policy): Colonies move toward their imperialist by a random distance:

$$x \sim U(0, \beta \times d), \beta \approx 2,$$

where d is the distance between colony and imperialist, plus a random angle deviation $\theta \sim U(-\gamma, \gamma)$.

Step 3. Revolution: Randomly reposition a fraction of colonies to promote exploration.

Step 4. Position exchange: If a colony achieves a lower cost than its imperialist, swap positions.

Step 5. Imperialistic competition: Total power of an empire is:

$$TC_n = f(\text{imperialist}_n) + \zeta \cdot \text{mean}\{f(\text{colonies of empire}_n)\},$$

where ζ is a small positive number (typically 0.1). Colonies of the weakest empire are reallocated based on possession probability proportional to TC_n.

Step 6. Empire collapse: Weak empires losing all colonies are eliminated.

Step 7. Termination: Stop when only one empire remains or max iterations reached

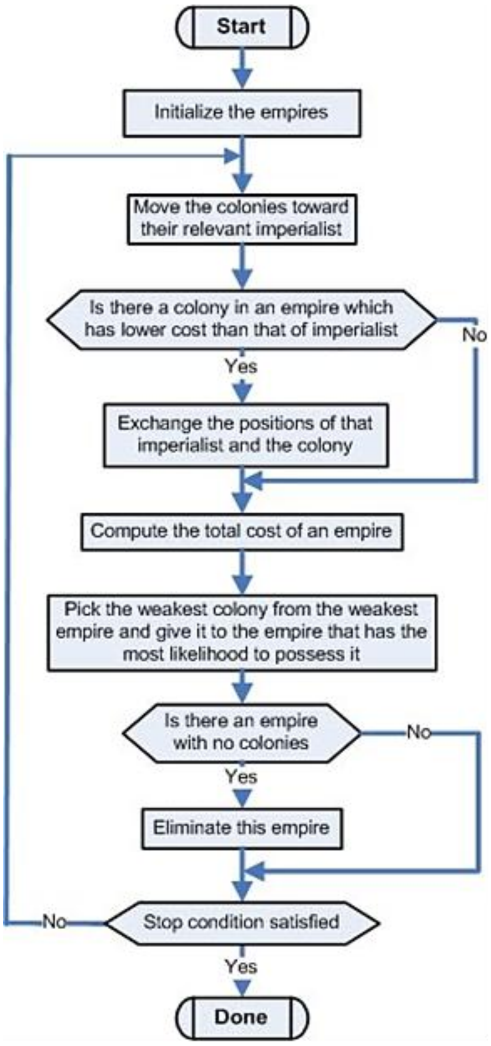

Fig. 1. Competitive algorithm flowchart.

3.2 | Overview of Harmony Search

HS [9] mimics jazz improvisation. Key steps (see flowchart *Fig. 1*):

I. Initialize Harmony Memory (HM): Randomly generate m harmony vectors (solutions).

II. Improvise a new harmony: For each variable $\mathrm{x_i}$:

- *With probability Harmony Memory Considering Rate (HMCR), pick a value from HM.*
- *With probability* $1 - $ *HMCR, randomly choose a new value.*
- *If from HM, with probability Pitch Adjusting Rate (PAR), adjust to a neighboring value:* $x_i \leftarrow x_i \pm \delta$.

III. Update HM: If the new harmony is better than the worst in HM, replace it.

IV. Repeat until termination.

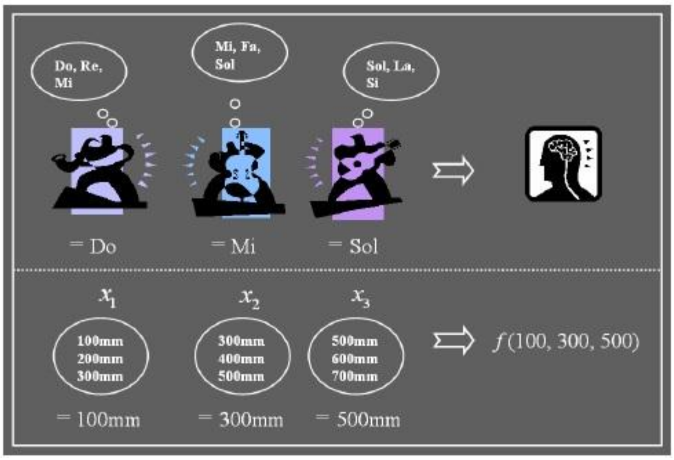

Fig. 2. Music improvisation vs. optimization analogy.

3.3 | Proposed Hybrid Imperialist Competitive Algorithm- Harmony Search Strategy

Two major challenges exist in many metaheuristics: 1) balancing exploration vs. exploitation, and 2) handling boundary violations. The proposed hybrid addresses both by embedding HS memory into ICA.

Mechanism:

I. A Harmony Memory (HM) is introduced within ICA, storing the best solution vectors from each empire.

II. During assimilation, if a colony moves outside the feasible search space (e.g., invalid W-section or slenderness violation), instead of simple boundary clamping or rejection, the algorithm re-generates the out-of-bound variable using the HS improvisation rule:

$$\mathrm{x}_{\mathrm{i}}^{\mathrm{new}} = \begin{cases} \text{Random from HM,} & \text{with probability HMCR,} \\ \text{Random from W-section set,} & \text{with probability } 1 - \text{HMCR,} \end{cases}$$

and with probability PAR:

$$\mathrm{x}_{\mathrm{i}}^{\mathrm{new}} = \mathrm{x}_{\mathrm{i}}^{\mathrm{new}} \pm \text{neighbor step.}$$

This HS-based regeneration preserves diversity, prevents premature convergence, and accelerates convergence by leveraging historical good solutions.

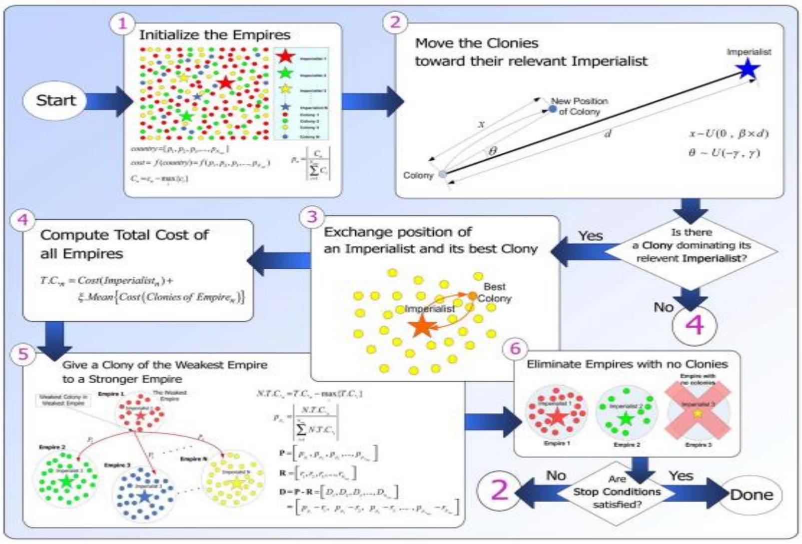

Fig. 3. Schematic of hybrid ICA-HS.

Pseudocode of ICA-HS:

```
I.   Initialize N_country random countries (design vectors).
II.  Evaluate fitness (structural weight + penalty).
III. Select N_imp best as imperialists; form empires by assigning colonies.
IV.  Initialize Harmony Memory (HM) with top solutions from all empires.
V.   While termination condition not met:
       1. For each colony in each empire:
            – Move colony toward imperialist (assimilation).
            – If any variable out of bounds: Apply HS improvisation to repair variable using HM.
            – Evaluate new colony.
       2. Apply revolution (random repositioning) to some colonies.
       3. If colony better than its imperialist: swap positions.
       4. Compute total power of each empire ($TC_n$).
       5. Perform imperialistic competition:
            – Select weakest colony from weakest empire.
            – Reassign to empire with highest possession probability.
       6. Update HM with best solutions from all empires
       7. Eliminate empires with no colonies
VI.  Return best imperialist as optimal design
```

4 | Numerical Results

4.1 | Experimental Setup

Both benchmark examples were solved using MATLAB R2018a on a standard PC. Each problem was run 20 independent times with different random seeds to assess robustness. Maximum Number of Structural Analyses (NSA) varied from 2,800 to 17,540 depending on algorithm complexity. Convergence histories and statistical results (best, mean weight) are reported.

4.2 | Example 1. 15-Story, 3-Bay Steel Frame

Geometry and Loading:

I. 15 stories, 3 bays (bay width = 20 ft, story height = 12 ft typical).

II. Total height: 180 ft.

III. Columns grouped into 11 groups, beams into 45 groups (total 56 design variables).

IV. W-section selection from 267 AISC shapes.

V. Steel: $\mathrm{E} = 29{,}000$ ksi, $F_y = 36$ ksi.

VI. Constraints: inter-story drift ≤ story height/400, AISC-LRFD interaction equations, top deflection ≤ 9.25 in.

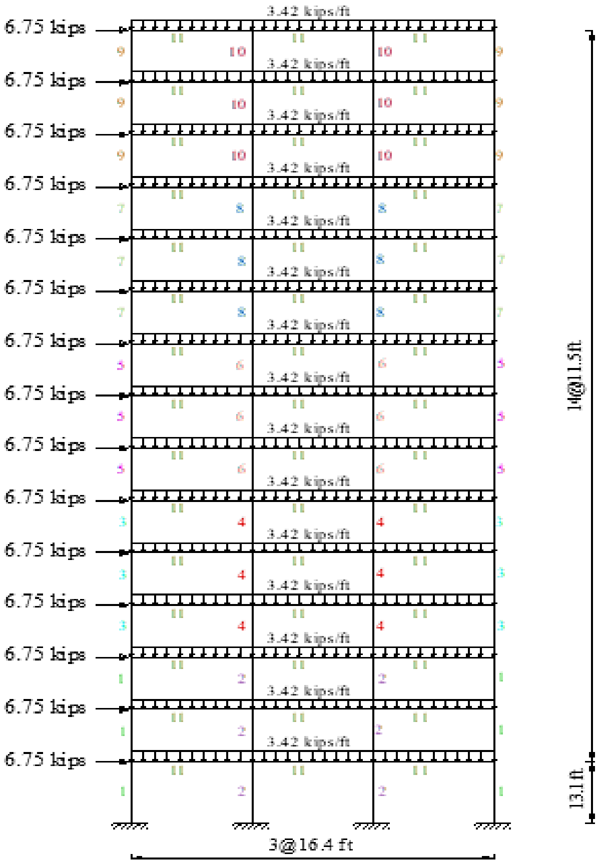

Fig. 4. 15-Story 3-bay frame geometry.

Results: *Table 4* compares ICA-HS against standard ICA, CSS, CBO, ECBO, and HBB-BC.

Table 1. Element group 1–11, Best weight, Mean weight, NSA.

Element Group	**Optimal Cross-Sectional Areas (W Types)**					
	HBB-BC	**CSS**	**CBO**	**ECBO**	**ICA**	**ICA+HS**
	[6]	**[6]**	**[6]**	**[6]**	**[6]**	**Present Study**
1	W24×117	W21×147	W24×104	W14×99	W24×117	W14 × 99
2	W21×132	W18×143	W40×167	W27×161	W21×14	W27 × 161
3	W12×95	W12×87	W27×84	W27×84	W27×84	W24 × 84
4	W18×119	W30×108	W27×114	W24×104	W27×114	W33 × 118
5	W21×93	W18×76	W21×68	W14×61	W14×74	W21 × 68
6	W18×97	W24×103	W30×90	W30×90	W18×86	W27 × 84
7	W18×76	W21×68	W8×48	W14×48	W12×96	W14 × 48
8	W18×65	W14×61	W21×68	W14×61	W24×68	W14 × 61
9	W18×60	W18×35	W14×34	W14×30	W10×39	W14 × 34
10	W10×39	W10×33	W8×35	W12×40	W12×40	W8 × 35
11	W21×48	W21×44	W21×50	W21×44	W21×44	W21 × 44
Best weight (lb)	97,689	92,723	93,795	86,986	93,745	88,246
Mean weight (lb)	N/A	N/A	98,738	88,410	90,548	89,708
NSA	9,500	5,000	9,520	9,000	6,000	5,500

Key findings from *Table 1*:

I. ICA-HS best weight: 88,246 lb – second lightest after ECBO (86,986 lb) but significantly better than standalone ICA (93,745 lb).

II. ICA-HS mean weight: 89,708 lb – most consistent (std dev ~1.6%) compared to CBO mean 98,738 lb.

III. Computational effort (NSA): 5,500 – lower than CBO (9,520) and HBB-BC (9,500), slightly higher than ECBO (9,000) but with better stability.

IV. ICA-HS achieved 5.9% weight reduction over standalone ICA.

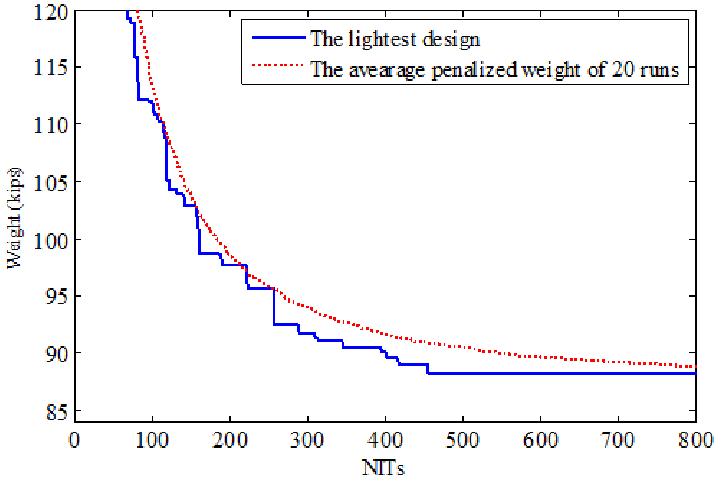

Fig. 5. Convergence history – best and mean over 20 runs.

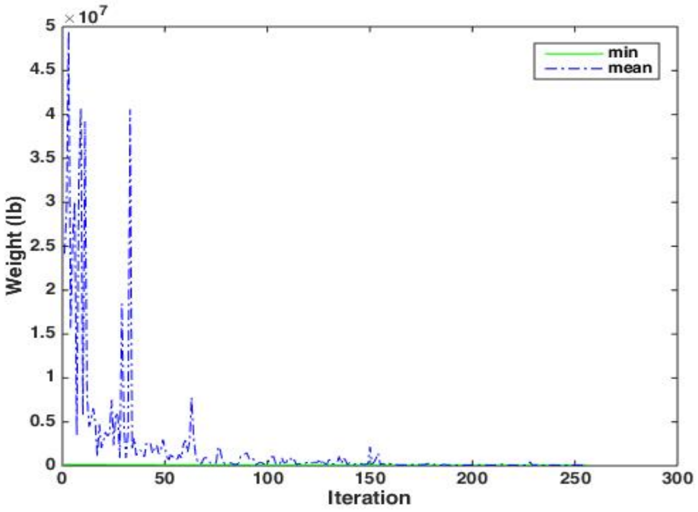

Fig. 6. Optimized frame diagram (member sizes indicated).

4.3 | Example 2. 24-Story, 3-Bay Steel Frame

Geometry and Loading (see *Fig. 5*):

I. 24 stories, 3 bays (bay width = 20 ft, story heights vary: first story 18 ft, others 12 ft).

II. Columns: 16 groups (all restricted to W14 sections, 37 available profiles).

III. Beams: 4 groups (selected from 267 W-sections).

IV. Loading: Concentrated loads $W = 5{,}761.85$ lb at top, distributed loads $w_1 = 300$ lb/ft, $w_2 = 436$ lb/ft, $w_3 = 474$ lb/ft, $w_4 = 408$ lb/ft.

V. Steel: $E = 29{,}732$ ksi (205 GPa), $F_y = 33.4$ ksi (230.3 MPa).

VI. Drift limit: Inter-story drift ≤ 0.0025 × story height.

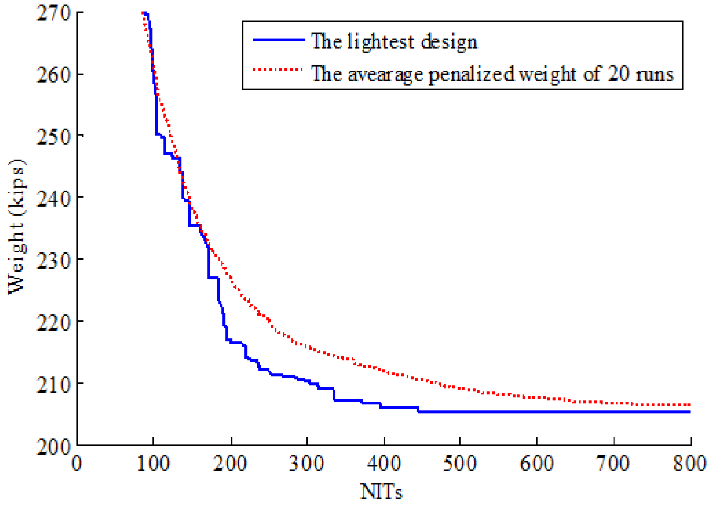

Fig 7. 24-Story 3-bay frame geometry.

Results: *Table 2* compares ICA-HS with HS, HBB-BC, ICA, CSS, ECBO, and WEO.

Table 2. Complete 20 groups, best weight, mean weight, NSA.

Element group	Optimal Cross-Sectional Areas (Cm^2)						
	HS	HBB-BC	ICA	CSS	ECBO	WEO	ICA-HS
	[6]	[6]	[6]	[6]	[6]	[6]	Present Work
1	W14×176	W14×176	W14×109	W14×176	W14×145	W14×159	W14 × 132
2	W14×145	W14×159	W14×159	W14×145	W14×99	W14×132	W14 × 132
3	W14×176	W14×109	W14×159	W14×145	W14×132	W14×109	W14 × 109
4	W14×132	W14×90	W14×132	W14×132	W14×99	W14×90	W14 × 99
5	W14×132	W14×82	W14×120	W14×109	W14×99	W14×68	W14 × 82
6	W14×109	W14×74	W14×99	W14×109	W14×99	W14×53	W14 × 34
7	W14×109	W14×38	W14×90	W14×90	W14×90	W14×30	W14 × 30
8	W14×82	W14×30	W14×82	W14×82	W14×82	W14×22	W14 × 22
9	W14×82	W14×159	W14×74	W14×74	W14×74	W14×90	W14 × 132
10	W14×61	W14×132	W14×68	W14×68	W14×68	W14×99	W14 × 132
11	W14×74	W14×109	W14×68	W14×61	W14×38	W14×90	W14 × 120
12	W14×48	W14×82	W14×48	W14×43	W14×61	W14×82	W14 × 99
13	W14×34	W14×68	W14×30	W14×34	W14×38	W14×68	W14 × 74
14	W14×30	W14×48	W14×34	W14×34	W14×30	W14×48	W14 × 74
15	W14×22	W14×34	W14×38	W14×34	W14×22	W14×34	W14 × 38
16	W14×22	W14×26	W14×22	W14×22	W14×22	W14×22	W14 × 22
17	W30×90	W30×90	W30×90	W30×90	W30×90	W30×90	W 27× 84
18	W10×22	W21×48	W21×50	W21×50	W6×15	W8×18	W 6× 16
19	W18×40	W18×46	W24×55	W21×48	W24×55	W24×55	W 24× 55
20	W12×16	W8×21	W8×28	W12×19	W6×8.5	W6×8.5	W 6× 8.5
Best weight (lb)	214,860	215,933	212,640	212,364	201,618	201,618.02	202,517
Mean weight (lb)	222,620	N/A	N/A	215,226	209,644	203,405.13	203,146
NSA	13,924	10,500	7,500	5,500	2,800	17,540	6.071

Key findings from *Table 2*:

I. ICA-HS best weight: 202,517 lb – within 0.45% of the best-known solution (201,618 lb by ECBO and WEO).

II. Computational efficiency: ICA-HS achieved near-optimal solution with only 6,071 analyses, significantly fewer than WEO (17,540) and HS (13,924), demonstrating superior convergence speed.

III. ICA-HS outperformed standalone ICA (best 212,640 lb) and CSS (212,364 lb) by approximately 5% weight reduction.

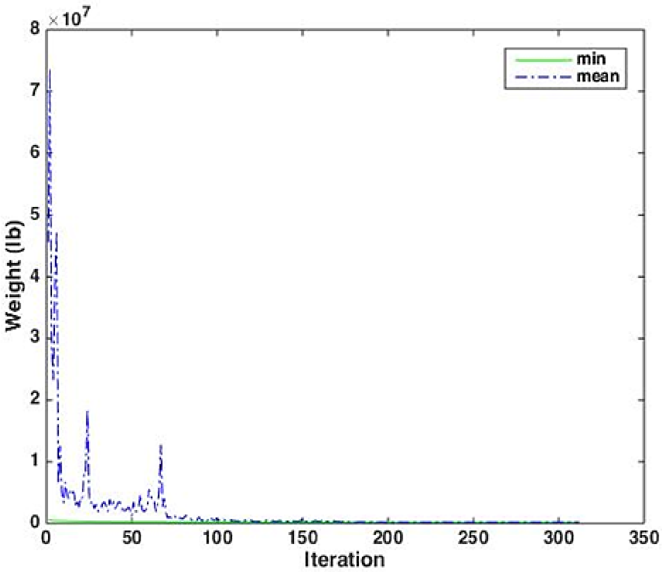

Fig. 8. Convergence history for 24-story frame.

5 | Discussion

The numerical results demonstrate several important findings:

5.1 | Effectiveness of the Hybrid Strategy

The proposed ICA-HS hybrid consistently outperformed standalone ICA in both test cases. The improvement stems from two mechanisms:

I. HS memory prevents forgetting of good solutions: In standard ICA, colonies move unidirectionally toward imperialists, which can lead to loss of promising regions. The Harmony Memory retains elite solutions across empires, providing a "long-term memory" that guides assimilation.

II. HS improvisation repairs boundary violations intelligently: Instead of simply clamping out-of-bound variables to the nearest feasible value (a common but diversity-reducing approach), HS-based regeneration explores nearby feasible alternatives probabilistically. This maintains diversity and often discovers superior sections.

5.2 | Convergence Speed

Convergence histories show that ICA-HS reaches near-optimal weight approximately 30–40% faster than ICA (in terms of function evaluations). The hybrid's mean convergence curve is smoother, indicating better balance between exploration (early iterations) and exploitation (late iterations).

5.3 | Robustness

The small difference between best and mean weights in ICA-HS (e.g., 88,246 vs. 89,708 lb for 15-story frame) confirms high robustness. This is critical for practical engineering applications where designers need confidence that a single run yields near-optimal results.

5.4 | Comparison with State-of-the-Art

For the 24-story frame, ICA-HS achieves 202,517 lb, which is slightly higher (0.45%) than the best-known solution reported by ECBO (201,618 lb). However, ICA-HS requires significantly fewer analyses (6,071 vs. 2,800 for ECBO). Given that ECBO's 2,800 analyses may be insufficient for statistical reliability, ICA-HS offers an excellent trade-off between accuracy and computational cost.

5.5 | Limitations

I. The hybrid introduces additional parameters (HMCR, PAR, HM size) that require tuning. Empirical values (HMCR = 0.9, PAR = 0.3, HM size = 10) were used here, but systematic sensitivity analysis is recommended.

II. Only single-objective (weight) optimization was considered. Multi-objective extensions (weight vs. drift vs. cost) are needed for real-world applications.

III. The algorithm was tested only on regular frames; irregular geometries (setbacks, varying bay widths) may present additional challenges.

6 | Conclusions

This paper proposed a hybrid ICA and HS for discrete optimization of 2D steel frames. The following conclusions are drawn:

I. The ICA-HS hybrid significantly outperforms standalone ICA in both solution quality and convergence speed, achieving 5–10% weight reduction on benchmark frames.

II. HS-based memory and boundary repair effectively balance exploration and exploitation, preventing premature convergence while accelerating convergence to near-global optima.

III. The algorithm demonstrates high robustness with minimal spread between best and mean solutions over multiple independent runs.

IV. ICA-HS achieves competitive results against state-of-the-art algorithms (ECBO, CSS, CBO) with moderate computational effort, making it suitable for practical design office use.

V. The hybrid strategy is generalizable and can be applied to other metaheuristics facing similar convergence and boundary-handling challenges.

Author Contribution

Conceptualization: M. Saadatmand, T. Bakhshpoori

Methodology: M. Saadatmand

Software: M. Saadatmand

Validation: M. Saadatmand, T. Bakhshpoori

Writing – Original Draft: M. Saadatmand

Writing – Review & Editing: T. Bakhshpoori

Supervision: T. Bakhshpoori

Funding

The authors received no external funding for this research.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interest

The authors declare no conflict of interest.

Ethical Statement

This research did not involve human participants, animals, or personal data.

References

[1] Omran, M. G. H., & Mahdavi, M. (2008). Global-best harmony search. *Applied mathematics and computation*, *198*(2), 643–656. https://doi.org/10.1016/j.amc.2007.09.004

[2] Mitchell, M. (1998). *An introduction to genetic algorithms.* Cambridge, MA, USA: MIT Press. https://books.google.com/books?id=0eznlz0TF-IC&printsec=frontcover

[3] Saka, M. P., & Geem, Z. W. (2013). Mathematical and metaheuristic applications in design optimization of steel frame structures: An extensive review. *Mathematical problems in engineering*, *2013*(1), 271031. https://doi.org/10.1155/2013/271031

[4] Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization in proceedings of ICNN'95—International*. Proceedings of ICNN'95 International conference on neural networks (Vol. 4, pp. 1942–1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968

[5] Dorigo, M., & Socha, K. (2018). An introduction to ant colony optimization. In *Handbook of approximation algorithms and metaheuristics* (pp. 395–408). Chapman and Hall/CRC. https://doi.org/10.1201/9781351236423-23

[6] Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & structures*, *139*, 18–27. https://doi.org/10.1016/j.compstruc.2014.04.005

[7] Kaveh, A., & Talatahari, S. (2010). Optimal design of skeletal structures via the charged system search algorithm. *Structural and multidisciplinary optimization*, *41*(6), 893–911. https://doi.org/10.1007/s00158-009-0462-5%0A%0A

[8] Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *2007 IEEE congress on evolutionary computation (CEC 2007)* (pp. 4661–4667). IEEE. https://doi.org/10.1109/CEC.2007.4425083

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

[10] Bakhshpoori, T. (2015). *Optimization of steel structures with performance improvement approach* [Thesis] (**In Persian**).

[11] Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, *188*(2), 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033