A Novel Hybrid Size-Shape Framework for Multi-Objective Optimization of Reinforced Concrete Frames

Jaemni So[1], Seungjae Lee[1], Koichi Kusunoki[2], Donwoo Lee[1*]

[1] *School of Industrial Design & Architectural Engineering, Korea University of Technology & Education, 1600 Chungjeol-ro, Byeongcheon-myeon, Cheonan 31253, Chungcheongnam-do, Republic of Korea*

[2] *Earthquake Research Institute, The University of Tokyo, 1-1-1 Yayoi, Bunkyo-ku, Tokyo 113-0032, Japan*

Corresponding author: Donwoo Lee (lov1004ely@koreatech.ac.kr)

Abstract

This study introduces a transformative hybrid size-shape optimization framework that bridges the critical gap between theoretical structural optimization and practical, on-site constructability for reinforced concrete (RC) frames. The framework presents two core innovations to overcome the limitations of conventional methods. First, it decouples the computationally intensive serviceability limit state (SLS) checks from the main optimization loop by pre-calculating a project-specific, discrete section database, dramatically enhancing both computational efficiency and the practical applicability of the results. Second, it pioneers a Hybrid Size-Shape Optimization paradigm by treating 90-degree column rotation as an independent binary design variable, enabling intelligent and efficient management of structural stiffness—a strategy shown to improve optimization performance by over 9% compared to traditional approaches. Employing NSGA-II as the multi-objective optimization engine, the framework simultaneously addresses two conflicting objectives: minimizing the combined normalized total cost-embodied carbon indicator, which includes both economic and environmental factors, and minimizing the average stress ratio, indicative of structural conservativeness. Ultimately, this framework effectively derives a set of Pareto-optimal solutions that clearly illustrate the inverse trade-off between the two objectives, providing designers with a quantitative decision-making tool to navigate the trade-offs between cost-carbon emissions and structural conservatism, enabling informed decision-making that aligns with specific project goals.

Keywords: Reinforced Concrete, NSGA-II, Automatic Design, Hybrid Optimization, Practical Design, Shape Optimization

1. Introduction

While academic research in structural optimization has produced highly sophisticated algorithms, a persistent and critical gap remains between these theoretical models and the practical realities of the construction site. This challenge is particularly acute in the design of Reinforced Concrete (RC) frames, one of the most prevalent structural forms in contemporary architecture. The design of these structures is inherently a complex multi-objective optimization problem (MOOP), demanding the simultaneous consideration of numerous, often conflicting, objectives (Esfandiary M et al., 2016; Ehrgott, 2012). For decades, designers have navigated the core trade-off between minimizing construction costs for economic efficiency and ensuring structural safety against external forces like earthquakes (Djedoui N et al., 2025; Faghirnejad S, 2023). Recently, this complex balancing act has been further complicated by the urgent demand for carbon neutrality and Sustainable Design within the construction industry (Akadiri & Fadiya, 2013). Consequently, a truly practical and modern optimization approach must not only resolve the cost-safety dilemma but also holistically account for the embodied carbon produced during material production and construction processes (Paya-Zaforteza et al., 2009; Mergos P, 2024; Werner & Burns, 2012).

The traditional design methodology for RC structures predominantly relies on the designer's expertise and iterative trial and error. This traditional approach has inherent limitations in identifying optimal solutions that concurrently addresses multiple objectives and constraints with intricate trade-offs. As structures increase in size and complexity, the implementation of automated optimization techniques is becoming imperative. In this context, metaheuristic search techniques, such as Genetic Algorithms (GA), have gained recognition as powerful optimization tools capable of effectively identifying optimal solutions to complex and nonlinear problems like RC frames (Kaveh & Sabzi, 2011; Aslay S et al., 2024; Kaveh A et al., 2023; Dehnavipour H et al., 2019; Akin A & Saka M, 2015; Heydari F et al., 2025; Bekdaş & Nigdeli, 2014).

Over recent decades, research on the optimization of RC frame structures has exhibited significant progress. Initial investigations predominantly concentrated on single objectives, such as minimizing construction costs or structural weight (Govindaraj & Ramasamy, 2005; Mergos P, 2021; Bekdaş & Nigdeli, 2013). However, the contemporary research paradigm is increasingly oriented towards multi-objective optimization, which elucidates trade-offs between cost and performance, or between cost and environmental impact (Kaveh A et al., 2020). In addressing these multi-objective optimization challenges, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) has proven its efficacy in efficiently identifying the Pareto-optimal front in complex problems. This is achieved through features such as rapid non-dominated sorting, crowding distance operation to ensure solution diversity, and an elitism strategy for preserving superior solutions (Deb et al., 2002). Consequently, it has become the de facto standard algorithm (Babaei & Mollayi, 2016; Zavala et al., 2016; Nebro et al., 2022).

Given this escalating complexity, why do many current "optimal" solutions derived from research fail to bridge this gap and see direct implementation? The answer lies in prevalent methodological limitations that do not adequately address the practical requirements of construction sites. Firstly, numerous optimization studies have derived solutions by assuming design variables to be continuous, failing to account for the discrete nature of actual construction, such as standard rebar sizes or formwork units (Chutani & Singh, 2018; Gharehbaghi & Fadaee, 2012; Chaudhuri et al., 2021). Secondly, many studies define design variables like reinforcement ratio (ρ), which does not directly translate into feasible rebar detailing, potentially compromising the initial optimality during post-processing (Aga & Adam, 2015; Jiu-lin et al., 2020). Furthermore, most studies are confined to sizing optimization, neglecting the significant impact of shape variables—such as the orientation of column sections—on the overall structural stiffness, often by constraining column shapes or fixing their orientation based on initial assumptions (Kaveh A & Rezazadeh Ardebili S, 2021; Kaveh A et al., 2023; Mergos P, 2022; Esfandiari M et al., 2018). Lastly, the inclusion of complex and time-consuming serviceability limit state (SLS) reviews within the main optimization loop significantly increases computational costs, hindering practical application (Faghirnejad S, 2023; Kaveh A et al., 2023; Kaveh A & Ardebili S, 2023; Juliani & Gomes, 2021). This study directly confronts these limitations by proposing a novel framework designed for immediate practical application.

The primary aim of this study is to introduce a paradigm shift in the optimization of reinforced concrete frames, moving from theoretical abstraction to practical implementation. We achieve this through a novel multi-objective framework that ensures both computational efficiency and the direct constructability of its solutions. To realize this, we pursue the following specific objectives:

Firstly, the study seeks to maximize practicality and computational efficiency. A tailored discrete section database is developed to meet the design constraints of the given project, such as materials and section sizes. During this process, complex serviceability limit state checks, including crack control, are pre-processed. This approach separates calculation-intensive reviews from the main search loop, thereby enhancing optimization efficiency and ensuring the practicality of the final solution.

Secondly, this study pioneers a new Hybrid Size-Shape Optimization paradigm. Moving beyond simple sizing optimization, we introduce the 90-degree rotation of the column section as a key design variable. This novel approach facilitates intelligent multi-axis stiffness control and, as our results will demonstrate, achieves superior optimization performance compared to conventional methods that simply expand the search database.

Thirdly, the study verifies the framework's effectiveness. By surpassing the limitations of existing studies, the effectiveness of the proposed framework is validated through the adoption of buildings with diverse span lengths and irregular floor plans, which resemble actual building forms, as example problems.

Fourthly, the study provides quantitative decision support. It performs multi-objective optimization that simultaneously aims to minimize two conflicting objectives: 'the sum of normalized total cost and embodied carbon' as a comprehensive indicator of economic and environmental performance, and the 'minimization of average stress ratio' as an indicator of structural conservativeness. Utilizing the NSGA-II algorithm, the Pareto optimal front representing the trade-off between these two objectives is derived, illustrating the potential for designers to make informed decisions that balance economic, environmental, and structural safety considerations based on quantitative evidence.

This study examines three-dimensional reinforced concrete moment-resisting frames, with the design space delineated by a pre-established discrete section database. The optimization process employs DEAP (Distributed Evolutionary Algorithms in Python), a Python-based evolutionary computation library, to implement NSGA-II. The structural performance of each candidate design is assessed using OpenSees (Open System for Earthquake Engineering Simulation), a widely validated open-source structural analysis framework (McKenna, 1997). Based on the outcomes of this process, constraints related to strength (stress ratio), serviceability (deflection, interstory drift ratio, top floor displacement), and constructability (upper-lower column hierarchy) are evaluated.

2. Theoretical Background

2.1. Multi-Objective Optimization

MOOP pertains to the mathematical process of concurrently optimizing two or more conflicting objective functions. Numerous real-world challenges in disciplines such as engineering, economics, and the natural sciences exhibit complexities that cannot be adequately assessed using a singular criterion. For instance, in structural design, it is imperative to minimize costs while simultaneously maximizing structural robustness. In these scenarios, each objective typically exists in a trade-off relationship with the others, whereby enhancing one objective may result in the deterioration of another (Marler & Arora, 2004). Consequently, the aim of multi-objective optimization is not to identify a singular 'optimal solution,' but rather to investigate a set of viable solutions that effectively balance the various objectives.

MOOP can generally be mathematically formulated as follows (Coello Coello, 2006):

Minimize:

$$F(x) = [f_1(x), f_2(x), \cdots, f_k(x)]^T \tag{1}$$

Subject to:

$$g_j(x) \leq 0, \quad j = 1, 2, \ldots, m \tag{2}$$

$$h_l(x) = 0, \quad l = 1, 2, \ldots, p \tag{3}$$

$$x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \ldots, n \tag{4}$$

Here, $x = [x_1, x_2, \ldots, x_n]^T$ is a vector composed of decision variables. $F(x)$ is a vector consisting of objective functions, and the space in which this vector exists is called the objective space. $g_j(x)$ and $h_l(x)$ represent inequality and equality constraints, respectively. x_i^L and x_i^U refer to the boundary conditions of the decision variables. The feasible design space X, which is the set of decision variable vectors that satisfy all of the above constraints, is defined as follows. In other words, X refers to the entire valid region where the decision variable vector x may exist.

$$X = \left\{x \in \mathbb{R}^n \,\middle|\, g_j(x) \leq 0, \forall j; \quad h_l(x) = 0, \forall l; \quad x_i^L \leq x_i \leq x_i^U, \forall i\right\} \tag{5}$$

In multi-objective optimization problems, it is common that there is no single solution that simultaneously optimizes all objective functions. Therefore, the concept of 'Pareto Optimality' proposed by Vilfredo Pareto is used to evaluate the quality of solutions (Zitzler & Thiele, 1999).

Here are the key concepts for understanding 'Pareto optimality':

1. Dominance: Given two solutions $x_A, x_B \in X$, solution x_A dominates solution x_B (denoted as $x_A \succ x_B$) if $f_i(x_A) \geq f_i(x_B)$ $\forall i \in \{1, \ldots, k\}$ and $f_i(x_A) > f_i(x_B)$ $\exists j \in \{1, \ldots, k\}\}$.

2. Non-dominated Solution: A solution that is not dominated by any other solution within the feasible design space X is called a non-dominated solution or Pareto optimal solution. In other words, improving any objective function value of a non-dominated solution necessarily requires sacrificing at least one other objective function value.

3. Pareto Optimal Set (P^*): The set of all possible non-dominated solutions, which exists in the decision variable space X.

$$P^* = \{x' \in X | \neg\exists x' \in X, \ F(x') \succ F(x)\} \tag{6}$$

4. Pareto Front (PF^*): The set of objective function values corresponding to solutions in the

Pareto optimal set, which exists in the objective function space.

$$PF^* = \{F(x) | x \in P^*\} \quad (7)$$

The Pareto front visually clarifies the relationships between conflicting objectives, making it a very useful tool for decision-makers to select a final solution from among various optimal alternatives based on the project's requirements and their own preferences. In a posteriori preference-based approaches like this study, an approximation of the Pareto front is first obtained through optimization algorithms, and then the decision-maker selects the final solution based on these results.

2.2. NSGA-II Algorithm

NSGA-II, a multi-objective genetic algorithm introduced by Deb et al. in 2002, remains one of the most extensively utilized and successful algorithms in the domain of multi-objective optimization (Deb et al., 2002). This algorithm addressed significant limitations of its predecessor, NSGA, including high computational complexity, absence of elitism, and the requirement for users to manually adjust parameters to preserve solution diversity. A proficient multi-objective evolutionary algorithm must concurrently achieve two potentially conflicting objectives: convergence, ensuring that the generated solutions closely approximate the true Pareto front, and diversity, ensuring that the solutions are widely and evenly distributed across the entire Pareto front. The design of NSGA-II is predicated on three principal mechanisms developed to address these dual challenges.

2.2.1. Fast Non-dominated Sorting

Fast non-dominated sorting is an algorithm that classifies a population into Pareto non-dominated fronts by efficiently sorting them according to their rank or front. This is an algorithm that dramatically improves computational efficiency, reducing the computational complexity from $O(MN^3)$ in the original NSGA to $O(MN^2)$, where M is the number of objective functions and N is the population size. The procedure is as follows:

1. For each solution p in the population, calculate the following two values:
   - n_p: the number of solutions that dominate solution p (domination count)
   - S_p: the set of solutions dominated by solution p
2. All solutions with $n_p = 0$ are not dominated by any other solution and thus belong to the first non-dominated front (F_1). These are the most superior solutions in the current population.
3. For each solution p in the first front (F_1), decrease the n_q value by 1 for all solutions q in the set S_p that solution p dominates.

4. If the n_q value of any solution q becomes 0, that solution belongs to the second non-dominated front (F_2).
5. This process is repeated until all solutions are assigned to their respective fronts.

Algorithm 1: Fast-Non-Dominated-Sort(P) (Deb et al., 2002)

For each $p \in P$	
$S_p = \emptyset$	
$n_p = 0$	
For each $q \in P$	
If $(p \prec q)$ **then**	If p dominates q
$S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
Else if $(q \prec p)$ **then**	
$n_p = n_p + 1$	Increment the domination counter of p
If $n_p = 0$ **then**	p belongs to the first front
$p_{rank} = 1$	
$F_1 = F_1 \cup \{p\}$	
$i = 1$	Initialize the front counter
While $F_i \neq \emptyset$	
$Q = \emptyset$	Used to store the members of the next front
For each $p \in F_i$	
For each $q \in S_p$	
$n_q = n_q - 1$	
If $n_q = 0$ **then**	q belongs to the next front
$q_{rank} = i + 1$	
$Q = Q \cup \{q\}$	
$i = i + 1$	
$F_i = Q$	

2.2.2. Crowding Distance and Comparison Operator for Diversity Preservation

NSGA-II uses a diversity preservation mechanism called Crowding Distance to maintain population diversity. The crowding distance represents how closely packed the solutions are around a particular solution within the same front, thereby indicating the uniqueness of that solution's location. This measure is used to compare solutions in the selection process.

When comparing any two solutions i and j in the population, both the non-dominated rank (i_{rank}) and the crowding distance ($i_{distance}$) are considered. The priority between two solutions i and j is determined by the following crowded comparison operator $<_n$:

$$i <_n j \qquad if\ (i_{rank} < j_{rank})\ or\ (i_{rank} = j_{rank}\ and\ i_{distance} > j_{distance}) \tag{8}$$

This operator ensures that solutions with better non-dominated ranks are selected preferentially, and when two solutions have the same rank, the solution with a larger crowding distance is selected to maintain diversity. The $i_{distance}$ is calculated as follows:

1. Initialize the crowding distance of each solution $i = |F|$ in front F to 0.
2. For each objective function $m \in \{1, \dots, M\}$, perform the following steps:
   - Sort all solutions in front F in ascending order according to objective function f_m.
   - Assign infinite crowding distances to the boundary solutions in the sorted front:

$$d_1 = d_l = \infty \tag{9}$$

   - For the remaining solutions, calculate the crowding distance using the following formula:

$$d_j = d_j + \frac{f_m(j+1) - f_m(j-1)}{f_m^{max} - f_m^{min}}, \quad for\ j = 2, \dots, l-1 \tag{10}$$

Where d_j is the crowding distance of the j-th solution in the sorted front, $f_m(j)$ is the m-th objective function value of the j-th solution, and f_m^{max} and f_m^{min} are the maximum and minimum values of objective function f_m in the corresponding front, respectively.

Algorithm 2: Crowding-Distance-Assignment(I) (Deb et al., 2002)

$l = \lvert I \rvert$	number of solutions in I
For each i, set $I[i]_{distance} = 0$	initialize distance
For each objective m	
$I = sort(I, m)$	sort using each objective value
$I[1]_{distance} = I[l]_{distance} = \infty$	so that boundary points are always selected
For $i = 2$ to $(l-1)$	for all other points
$I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m)/(f_m^{max} - f_m^{min})$	

2.2.3. Generational Change Through Elitism

Elitism is an excellent method for finding optimal solutions or solutions close to optimal in evolutionary algorithms. NSGA-II uses a straightforward yet powerful elitism strategy as follows:

1. Combine the current population P_t of size N with the offspring population Q_t of size N generated through selection, crossover, and mutation operations to create a combined population R_t of size $2N$.

$$R_t = P_t \cup Q_t \tag{11}$$

2. Apply fast non-dominated sorting to the combined population R_t and create fronts $F_1, F_2, F_3, \dots$ in order.

3. Construct the new population P_{t+1} of size N by sequentially including solutions from the best fronts $F_1, F_2, F_3, \ldots$ until the population size is filled.

4. If including all solutions from a particular front F_k would exceed the population size N, only a portion of the solutions from front F_k should be included. The solutions within front F_k are selected based on the crowded comparison operator ($<_n$), prioritizing solutions with larger crowding distances. This ensures that solutions closer to the optimal P_{t+1} are preserved.

This approach ensures that the current generation and all offspring compete fairly, with only the most superior N solutions being selected for the next generation, thereby significantly improving the convergence and diversity characteristics of the algorithm.

Algorithm 3: Generational Change Through Elitism (Deb et al., 2002)

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} =$ fast-non-dominated-sort(R_t)	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
Until $\lvert P_{t+1} \rvert + \lvert \mathcal{F}_i \rvert \leq N$	until the parent population is filled
Crowding-distance-assignment($\mathcal{F}_i$)	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
Sort($\mathcal{F}_i, \prec_n$)	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1:(N - \lvert P_{t+1} \rvert)]$	choose the first $(N - \lvert P_{t+1} \rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} =$ make-new-pop(P_{t+1})	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

2.3. Structural Design Constraints and Penalty Function

The constraints that an optimization algorithm must adhere to are delineated by the design philosophy and codes of structural engineering. The optimization framework employed in this study is grounded in the Strength Design Method (SDM), which constitutes the basis of contemporary reinforced concrete design. The Strength Design Method, also referred to as the Limit State Design (LSD), was developed to address the limitations inherent in the Allowable Stress Design (WSD) initially utilized. Unlike WSD, which presumes that materials behave elastically under service loads, SDM concentrates on the ultimate state leading to structural failure, thereby ensuring a more rational and consistent level of safety (MacGregor, 1997). The fundamental principle of the Strength Design Method is to ensure that the design strength, defined as the maximum capacity a member can resist, is greater than or equal to the required strength imposed by external loads. This is articulated through the following basic inequality.

$$\emptyset R_n \geq U \qquad (12)$$

U refers to the required strength, which denotes the load effects (such as bending moment, shear force, etc.) generated in a member by the factored load. This factored load is obtained by multiplying various loads, such as dead load (D) and live load (L), by their respective load factors (γ), and subsequently combining them. Load factors are assigned values greater than 1.0 to account for uncertainties and prediction errors associated with the loads. R_n represents the nominal strength, which is the theoretical strength of the member, calculated using the nominal dimensions and specified material strengths. ϕ is the strength reduction factor, which reduces the nominal strength to account for material strength variability, construction errors, and uncertainties in cross-sectional dimensions. This factor is less than 1.0 and varies according to the ductility of the failure mode: a higher value is applied when ductile failure is anticipated, such as in tension-controlled sections, and a lower value is applied when brittle failure is possible, such as in compression-controlled sections, to enhance the safety of structures.

This approach rationally addresses the uncertainties inherent in both loads and material strengths with separate safety factors—load factors and strength reduction factors—and provides a consistent level of reliability under various loading conditions. Strength design methods define various limit states where a structure ceases to meet its intended function and aim to control the probability of reaching each limit state to an acceptable level or lower. Limit states are broadly classified into ultimate limit states for safety, and serviceability limit states for function and comfort; in optimization problems, each is transformed into a constraint function of the form $g(x) \leq 0$.

- Ultimate Limit States (ULS) pertain to failure conditions that are directly associated with life safety, such as the structural collapse. The evaluation of ULS involves verifying that the flexural strength, shear strength, and axial force-moment interactions of all structural members surpass the required strengths under factored loads, in accordance with ACI 318 standards. Additionally, it necessitates compliance with the minimum and maximum reinforcement ratio requirements to promote ductile behavior.

- Serviceability Limit State (SLS) refers to conditions related to the operational functionality of a structure and the comfort of its occupants. Critical aspects for evaluation include the regulation of excessive deflection and crack width under service loads, as delineated in ACI 318 standards. Additionally, it may encompass interstory drift and top-story displacement limits in response to lateral loads, such as those induced by wind or seismic activity, as specified in ASCE 7 (ASCE, 2016).

In this study, the principles of strength design methods serve as critical constraints that each candidate design evaluated by the optimization algorithm must satisfy. Specifically, the design strength (ϕR_n) of all members must exceed the required strength (U), and serviceability limits, such as deflection, must also be adhered to. Table 1 summarizes these limit states and their corresponding constraint formulations.

Table 1. Summary of Structural Design Limit States and Constraints

Category	Limit State	Objective	Constraint Formulation	Key Checks	Governing Code
Safety	ULS	Prevention of destruction such as collapse, overturning, and buckling of structures	$g_{ULS}(x) = \frac{U}{\emptyset R_n} - 1 \leq 0$ $g_{ULS,rebar}(x) = \begin{cases} \frac{\rho_{min}}{\rho} - 1 \leq 0 \\ \frac{\rho}{\rho_{max}} - 1 \leq 0 \end{cases}$	Flexural strength, shear strength, axial force-moment correlation (P-M), maximum/minimum reinforcement ratio	ACI 318
Serviceability	SLS	Ensuring functionality and comfort by preventing excessive deformation, cracks, and vibrations	$g_{SLS,defl}(x) = \frac{\delta_{actual}}{\delta_{allowable}} - 1 \leq 0$ $g_{SLS,drift}(x) = \frac{\Delta_{actual}}{\Delta_{allowable}} - 1 \leq 0$ $g_{SLS,crack}(x) = \frac{\omega_{actual}}{\omega_{allowable}} - 1 \leq 0$ $g_{SLS,top}(x) = \frac{\Delta_{top}}{H/400} - 1 \leq 0$	Deflection, inter-story displacement, top floor displacement, crack width control	ACI 318, ASCE 7

3. Proposed Optimization Framework

The optimization framework introduced in this study comprises two primary stages: (1) the construction of a project-specific database and (2) the optimization process, as illustrated in Figure 1. This bifurcated approach delineates the search space for a particular design problem and segregates the computationally demanding individual section review from the principal optimization loop, thereby enhancing both efficiency and practicality.

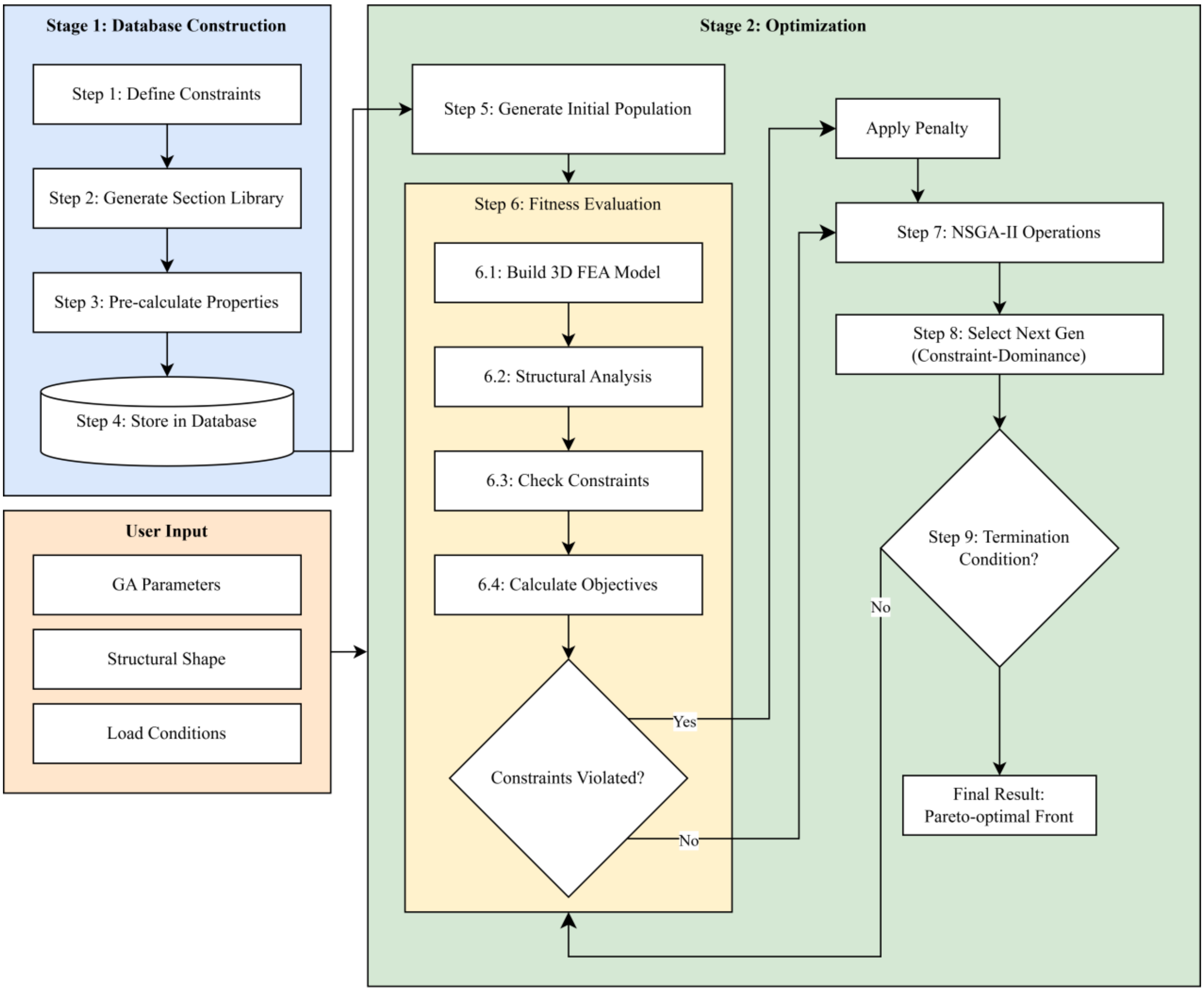


Figure 1. Overall Flowchart of the Proposed Optimization Framework

3.1. Overall Optimization Process

The initial phase of database construction involves a preparatory procedure that delineates a tailored search space for a specific optimization problem. By inputting particular constraints pertinent to a given project—such as materials, cross-sectional dimensions, and rebar specifications (Figure 1, Step 1)—a library is generated comprising all feasible combinations of beam and column sections that meet these criteria (Step 2). For each cross-section, performance metrics (strength, serviceability) and attributes (cost, embodied carbon) are pre-calculated through individual structural analyses (Step 3), and the results are stored in a database for utilization in the optimization phase (Step 4).

The second optimization stage constitutes the central process that seeks optimal solutions utilizing the customized database generated in the preceding stage, employing the NSGA-II algorithm.

This algorithm initiates the process by forming the initial population through the combination of section IDs from the database and the rotation status of columns (Step 5). It then conducts repeated fitness evaluations (Steps 6–8) for each candidate design. This iterative process continues until the predefined termination conditions (Step 9) are satisfied, ultimately yielding the Pareto optimal front.

3.2. Project-Specific Section Database Generation

This framework initially establishes a tailored discrete section database that adheres to the practical constraints delineated for each optimization problem. By conceptualizing the optimization problem as a discrete combinatorial issue within a realistic search space, it ensures that the ultimate solution can be directly implemented without necessitating additional post-processing. Given the computationally demanding nature of database creation, this phase was executed using MATLAB for its robust numerical computation capabilities, whereas the primary framework, which conducts optimization utilizing the generated database, was implemented in a Python environment leveraging the DEAP and OpenSees libraries.

For the numerical example presented in this study, the search space was delineated by a comprehensive set of design parameters, including section dimensions, material strengths, and reinforcement details. The full scope of these parameters is detailed in Table 2. The number of potential unique sections, derived from the combinations of these parameters, is on the order of tens of thousands, making an exhaustive enumeration computationally prohibitive and unnecessary for a metaheuristic search. Therefore, a database of 1,000 sections each for beams and columns was strategically generated. This was achieved by applying random sampling to the defined parameters, thereby constructing a representative discretized search space that statistically approximates the distribution of the entire potential design space. This approach is based on assumption that a near-optimal solution can be constructed from combinations of these representative cross-sections, as it ensures the search space is sufficiently diverse to capture a wide spectrum of performance characteristics—from highly economical to robustly conservative. The actual costs and embodied carbon amounts corresponding to the generated cross-section IDs are illustrated in Figure 2, and Table 3 summarizes the unit values used to calculate these properties.

The generated database is organized based on key performance indicators to enhance the algorithm's search efficiency and ensure consistency in result analysis. Beam cross-sections are arranged in ascending order according to the major axis design flexural strength (ϕM_n), while column cross-sections are sorted in ascending order based on the area of the axial force-moment interaction diagram (P-M Diagram) of the major axis. Consequently, within each database, a lower section ID signifies lower section performance and cost.

3.2.1. Beam Section Database Generation

Given that flexural and shear resistance are critical in beams, the design flexural strength (ϕM_n) and design shear strength (ϕV_n) are computed as primary performance indicators. The generated code initially calculates the balanced reinforcement ratio (ρ_b) in accordance with ACI 318 regulations. Based on this calculation, it randomly determines the tensile reinforcement ratio (ρ_s) and the compression reinforcement ratio (ρ_s') within a range that promotes ductile failure. Consequently, all beams included in the database are doubly reinforced with compression reinforcement. All cross-sections comply with the minimum and maximum reinforcement ratio, bar spacing, and cover thickness requirements, and they also fulfill serviceability requirements through crack width assessments.

3.2.2. Column Section Database Generation

Columns are designed to withstand axial force, biaxial bending, and shear simultaneously, with all strengths being calculated except for torsion. The generated code randomly determines the reinforcement quantity within the total reinforcement ratio (ρ_g) range specified by ACI 318 (1%~8%). The section is constrained to a rectangular shape, and after positioning four corner rebars as a basic arrangement, additional rebars are randomly distributed along the major or minor axis. This results in various patterns, including not only simple arrangements on all four sides but also configurations that enhance stiffness along a specific axis, such as two-side reinforcement. Through this methodology, axial force-moment interaction diagrams and biaxial shear strengths are computed and incorporated into the database.

Table 2. Parameters for Project-Specific Database Generation

Parameter	Beam	Column
Common		
Concrete Strength (f_{ck})	21, 27 MPa	21, 27 MPa
Rebar Strength (f_y)	400, 500 MPa	400, 500 MPa
Maximum Aggregate Size	25 mm	25 mm
Geometry		
Section Width (b)	200 ~ 600 mm (50mm inc.)	300 ~ 600 mm (50mm inc.)
Section Depth (h)	300 ~ 1500 mm (50mm inc.)	300 ~ 1200 mm (50mm inc.)
Reinforcement		
Reinforcement Type	Doubly Reinforced	2-sided & 4-sided patterns
Reinforcement Ratio (ρ)	Derived from ρ_b	$0.01 \leq \rho_g \leq 0.08$
Main Bar Diameter	D19, D22	D19, D22, D25

Stirrup / Tie Diameter	D10, D13	D10, D13

Table 3. Fixed Values of Calculating Cross-section Properties

Concrete Cost	80,000 won/m^3
Steel Cost	1,000,000 won/m^3
Concrete Densityl	2,400 kg/m^3
Steel Density	7850 kg/m^3
Embodied Carbon of Concrete	0.15 $kgCO_2e/kg$
Embodied Carbon of Steel	1.99 $kgCO_2e/kg$

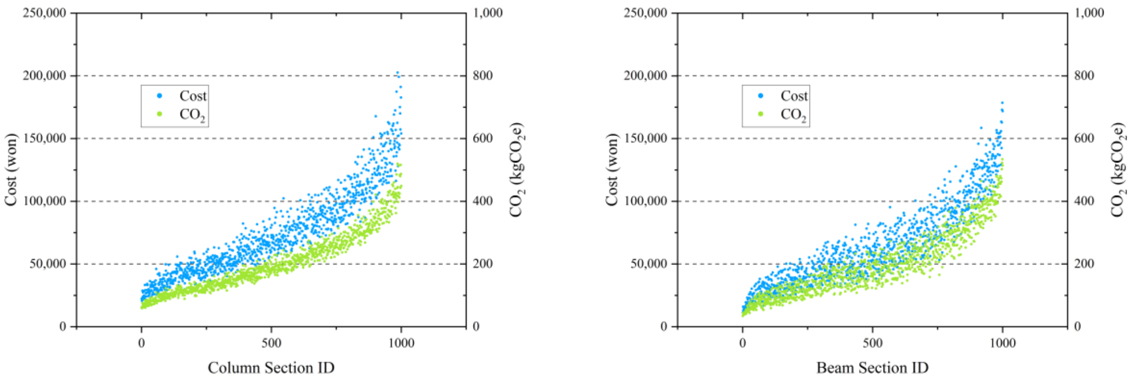


Figure 2. Distribution of Cost and Carbon by Cross-section ID

3.3. Optimization Methodology

The optimization methodology of this framework comprises decision of variables, a member grouping strategy for their efficient management, an objective function that delineates the optimization goal, and constraints to ensure a realistic design.

3.3.1. Grouping Strategy for Decision Variables

In practical construction settings, the process of grouping structurally analogous members—referred to as member grouping—by assigning them identical cross-sections is crucial for enhancing construction efficiency and minimizing formwork production costs. Furthermore, this grouping reduces the total number of variables in optimization problems, thereby facilitating more efficient algorithmic solution searches (Boscardin et al., 2019).

In this study, members with analogous structural functions were classified by location and floor to form groups. Columns are categorized based on their position within the plan (corner, perimeter, interior) and floor level (in units of two floors). Similarly, beams are classified according to their position within the plan (perimeter, interior) and floor levels in two-floor increments. The grouping of members is not considered a variable in the optimization process and is predetermined according to the method established by the user. Figure 3 illustrates the grouping strategy applied to the example problem in Section 4.1.

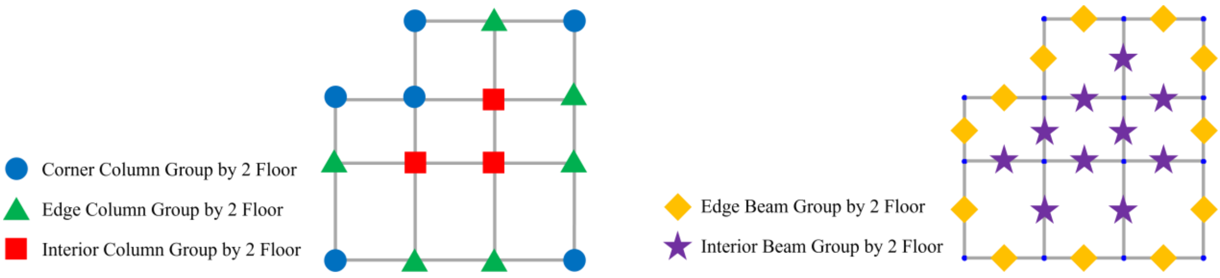


Figure 3. Group Strategy of Column and Beam

3.3.2. Decision Variables: Hybrid Sizing-Shape Optimization

In the optimization algorithm, each candidate design is represented by genetic information, referred to as chromosomes. The chromosome is structured as a vector of integers, which denote the unique identification numbers of the cross-sections assigned to each member group, as well as the rotation status of the column groups. The rotation status is encoded as binary integers, specifically 0 and 1. Figure 4 illustrates the chromosome of the four-story example building discussed in Section 4.1. This chromosome comprises a total of 16 genes, which include 6 column section IDs, 6 column rotation flags, and 4 beam section IDs.

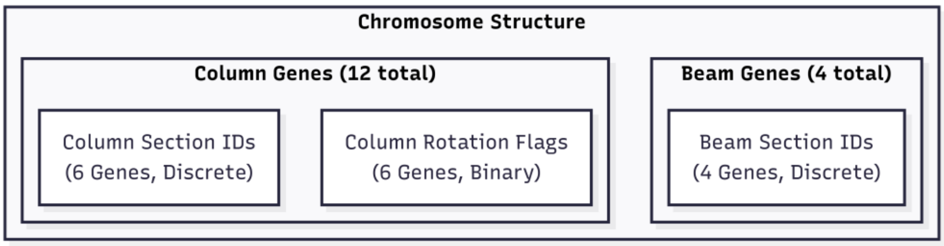


Figure 4. Composition of the Chromosome for the 4-Story Example

Thus, size optimization, which determines the dimensions of structural members through section identifiers, and shape optimization, which strategically manages the lateral stiffness of the structural system via column rotation indicators, are executed concurrently.

3.3.3. Objective Functions

This study conducts a multi-objective optimization to concurrently minimize two objectives, thereby examining the trade-off between economic efficiency and structural conservatism.

The first objective function, combined normalized total cost-embodied carbon indicator ($f_1(x)$), serves as a comprehensive indicator that evaluates both the total construction cost and the total embodied carbon. Given that these indicators are generally proportional to the material quantity and exhibit a strong linear correlation, they are amalgamated into a single cost indicator rather than being treated as independent objectives for optimization. At this juncture, due to the differing units and value ranges of the two indicators, each is normalized to the range [0, 1] prior to summation, to prevent the construction cost—which typically possesses larger numerical values—from overshadowing the embodied carbon indicator.

$$min f_1(x) = \omega_{cost}\left(\frac{Cost(x)-C_{min}}{C_{max}-C_{min}}\right) + \omega_{co2}\left(\frac{CO_2(x)-E_{min}}{E_{max}-E_{min}}\right) \tag{13}$$

In this context, $f_1(x)$ denotes the decision variable vector, while $Cost(x)$ and $CO_2(x)$ represent the total construction cost and total embodied carbon of the design, respectively. The parameters $C_{min/max}$ and $E_{min/max}$, employed for normalization, signify the theoretical minimum and maximum values derived from all section combinations within the database utilized for this optimization problem. The weight was consistently set to 1.0 across all cases in this study.

The second objective function, structural conservatism ($f_2(x)$), is characterized by the mean of the maximum stress ratios across all structural components. A reduced average stress ratio signifies a more conservative structural design.

$$min f_2(x) = Mean\ DCR(x) = \frac{1}{N_{members}} \sum_{i=1}^{N_{members}} DCR_{i,max}(x) \tag{14}$$

3.3.4. Constraints and Penalty Function

All candidate designs evaluated by the algorithm are required to comply with system-level constraints as delineated by the ACI 318 and ASCE 7 standards, thereby ensuring the safety, serviceability, and constructability of the structure. The primary constraints considered in this study are detailed in Table 4.

Table 4. Design Constraints

Category	Constraint Description	Formulation	Governing Code
Strength	DCR of any member	$\max_{i\in \text{members}}(DCR_i) \leq 1.0$	ACI 318
Serviceability	Inter-story Drift Ratio (Seismic Load)	$\Delta_{drift} \leq \Delta_{allowable,drift}$	ASCE 7
	Top-story Displacement (Wind Load)	$\Delta_{top} \leq H_{total}/400$	ASCE 7
	Beam Deflection (Simplified Check)	$h_{beam} \geq L_{beam}/21$	ACI 318

Constructability	Column Hierarchy	$Size_{uuper} \leq Size_{lower}$	Practical Requirement

This framework has adopted the Constraint-Dominance Principle to handle these constraints. To implement this principle, a normalized Constraint Violation (CV) score is calculated to quantify the degree to which each candidate design violates the constraints. The total violation score is computed by normalizing the violation ratio of each constraint and summing them up, as formulated in Equations 15-17.

$$CV(x) = \sum_{j=1}^{N_c} \omega_j \cdot \frac{M_j}{V_{j,max} - 1} \quad (15)$$

$$M_j = \max(0, V_j(x) - 1.0) \quad (16)$$

$$\omega_j = 1 \quad (17)$$

Here, $CV(x)$ is the total normalized constraint violation score for a given design solution represented by the decision vector x. The summation is performed over all N_c constraints (indexed by j). $V_j(x)$ represents the calculated violation ratio for the j-th constraint (i.e., actual value divided by the allowable limit), and M_j is the magnitude of this violation beyond the acceptable threshold of 1.0.

In this formulation, the explicit weights ω_j are set to unity (1.0). This is a deliberate choice to maintain objectivity and avoid introducing subjective bias in prioritizing constraints. However, this does not mean all violations are treated equally. A more sophisticated, implicit weighting mechanism is embedded within the normalization process itself. The normalization bounds, $V_{j,max}$, are not arbitrary values but are strategically established based on structural engineering principles, code philosophies, and damage assessment research, as detailed in Table 5. This approach provides crucial flexibility to the optimization algorithm while recognizing the relative severity of different violations. Crucially, preliminary validation studies confirmed that solutions exceeding these established bounds consistently required major design revisions, thus verifying their appropriateness as effective and realistic optimization boundaries.

Table 5. Engineering-Justified Normalization Bounds for Constraint Violations

Constraint (j)	Max. Allowable Ratio ($V_{j,max}$)	Engineering Justification	Code/Literature Reference
Strength	2.0	The 2.0 limit reflects the principle that strength failures must be strictly controlled. Violations	Based on ACI 318 strength

		beyond 100% (a factor of 2.0) are considered indicative of a fundamental design flaw rather than a simple resizing issue, thus marking a hard boundary for optimization.	design philosophy
Inter-story Drift	2.5	This bound is informed by FEMA P-58 fragility functions, which indicate that drift ratios up to 2.5 times the code limit generally correspond to a repairable damage state, providing a rational upper limit for serviceability.	ASCE 7-16, FEMA P-58
Top-story Displacement	3.0	Overall building displacement has less direct impact on structural damage compared to inter-story effects. Higher tolerance reflects this lower criticality.	Chopra (2017), Dynamics of Structures
Beam Deflection	2.0	ACI 318 deflection limits are primarily governed by serviceability concerns (e.g., aesthetics, occupant comfort) rather than safety. A factor of 2.0 is permitted as a soft constraint boundary, acknowledging that these conservative limits can be slightly exceeded during optimization before triggering significant functional issues.	ACI 318-19, Section 7.3
Column Hierarchy	1.2	A tight tolerance of 1.2 is used to strictly enforce the strong-column weak-beam principle, which is critical for seismic safety. The 20% allowance accommodates the discrete nature of the section database without compromising this fundamental design philosophy.	ACI 318-19, Section 18.7.3

The $CV(x)$ value, calculated through this rigorous process, is then used to determine the superiority of two solutions during the selection process, according to the constraint-priority principle outlined in Algorithm 4. This principle always considers a feasible solution ($CV = 0$) to be superior to an infeasible one ($CV > 0$). If both solutions are infeasible, it prefers the solution with the smaller CV value, thereby guiding the search efficiently toward the feasible design space.

Algorithm 4: Constraint-Dominance-Compare(A, B)

$CV_A \leftarrow A.violation\_score$	constraint violation of A
$CV_B \leftarrow B.violation\_score$	constraint violation of B
If($CV_A = 0 \land CV_B > 0$) return A	feasible solution is preferred
Else if($CV_A > 0 \land CV_B = 0$) return B	feasible solution is preferred
Else if($CV_A < CV_B$) return A	smaller violation is better
Else if($CV_A > CV_B$) return B	smaller violation is better
Else return Pareto_Dominance_Compare(A, B)	both feasible or equal violation

3.4. NSGA-II Application

The optimization framework introduced in this study is executed by integrating DEAP, a Python-based evolutionary computation library, with OpenSees, an open-source finite element analysis framework. DEAP serves as a robust tool that flexibly defines the fundamental components of genetic algorithms, including individuals, populations, genetic operators, and the evolutionary engine. In this research, DEAP is employed to generate individuals with the chromosome structure delineated in Section 3.3.2 and to implement the core selection mechanism of NSGA-II. The structural performance of each candidate design is assessed using the extensively validated OpenSees. Within the optimization loop, upon provision of the genetic information (section ID, rotation flag) for each individual, the framework automatically generates an OpenSees script to construct a 3D analysis model of the design. Following the execution of static analyses for all load combinations, the framework extracts analysis results, such as member forces and nodal displacements, to compute the objective function values as outlined in Section 3.3.3 and to verify constraint compliance as described in Section 3.3.4.

By integrating the execution of evolutionary operations using DEAP with the evaluation of structural performance through OpenSees, the entire workflow is completed to autonomously explore the search space of reinforced concrete frame structures with complex constraints, thereby identifying the optimal solution.

4. Numerical Example & Result Analysis

4.1. Target Structure, Load Conditions, Section Constraints

The structure selected for evaluating the efficacy of this optimization framework is an eight-story, three-dimensional reinforced concrete moment-resisting frame, as depicted in Figure 5. The floor plan exhibits an irregular configuration with varying span lengths.

The structural analysis considers various load conditions, including dead load, live load, wind load, and seismic load, with load combinations applied in accordance with ACI 318 and ASCE 7 standards. The loads on the beams of each floor are arranged in a checkerboard pattern, ensuring that the loaded areas vary for each floor, thereby reflecting realistic structural design practices. Detailed depictions of the load-bearing areas for each floor are provided in Figure 6. Table 6 presents a summary of the load values utilized in this example. Additionally, the self-weight of all structural members is automatically accounted for during analysis, based on pre-calculated unit weights from the database. Lateral forces are distributed to each floor following an inverted triangular distribution derived from the base shear force and are subsequently evenly allocated to each node on the floor. The magnitude of the lateral load acting on each node is detailed in Table 7, with an illustration provided in Figure 7. The cross-sectional search space for optimization is confined to the beam and column database, which is generated according to the project constraints outlined in Table 2 of Section 3.2.

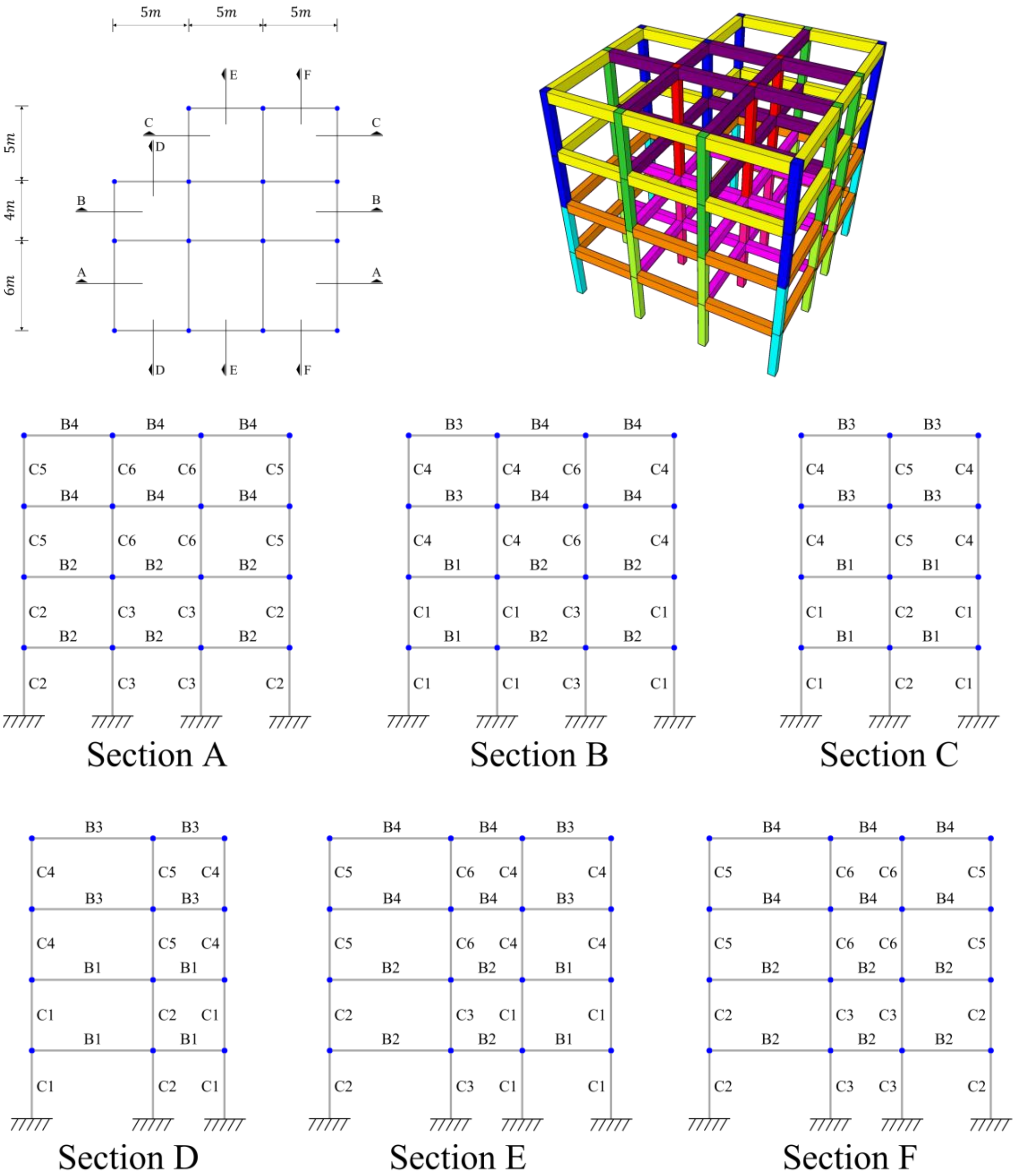


Figure 5. Geometrical Configuration of the Target Structure

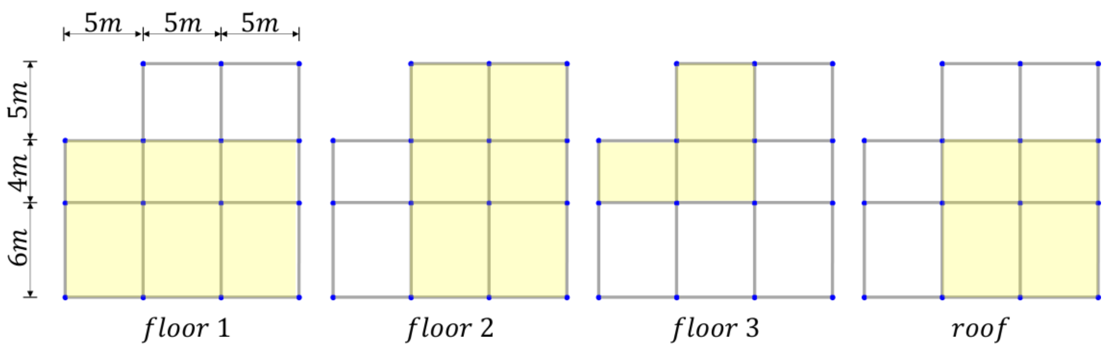


Figure 6. Checkerboard Load Pattern for Superimposed Gravity Loads per Floor

Table 6. Load Conditions for the Numerical Example

Load Type	Parameter	Value	Note
Gravity Loads	Superimposed Dead Load (DL)	25 kN/m	Checkerboard pattern
	Live Load (LL)	20 kN/m	Checkerboard pattern
Lateral Loads	Base Shear X (Seismic, Ex)	40 kN	Triangular distribution
	Base Shear Y (Seismic, Ey)	40 kN	Triangular distribution
	Base Shear X (Wind, Wx)	35 kN	Triangular distribution
	Base Shear Y (Wind, Wy)	38 kN	Triangular distribution

Table 7. Nodal Lateral Loads per Floor (kN)

Floor	Nodal Fx (Seismic)	Nodal Fy (Seismic)	Nodal Fx (Wind)	Nodal Fy (Wind)
1	0.267	0.267	0.233	0.253
2	0.533	0.533	0.467	0.507
3	0.8	0.8	0.7	0.76
roof	1.067	1.067	0.966	1.013

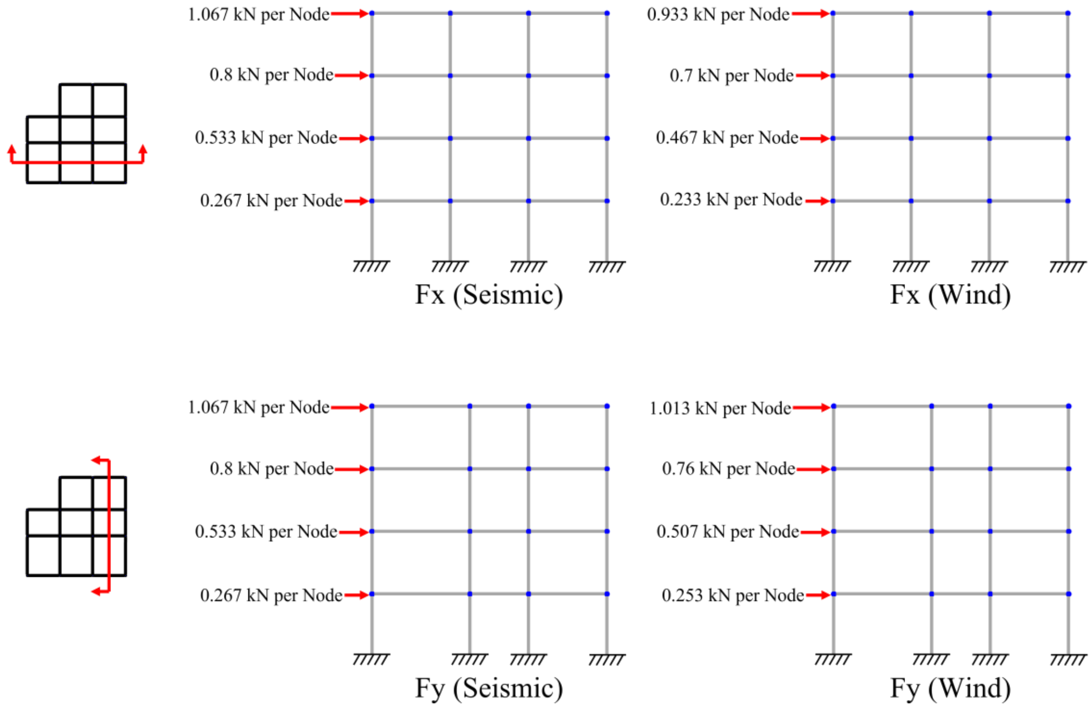


Figure 7. Graphically represented nodal lateral loads per floor (kN)

4.2. Determination of Optimal Genetic Algorithm Parameters

This section elucidates that the primary parameters of the genetic algorithm employed in the optimization experiments were selected through objective preliminary testing, and it offers a comprehensive account of the decision-making process. For parameter determination, the target structure maintained the same shape and loading area as the example problem, with only the load value being altered. The selection of optimal parameters was predicated on the convergence performance of the first objective function among the two objective functions. Separate experiments were conducted on the crossover strategy, parent selection strategy, and population to ascertain the optimal combination.

4.2.1. Crossover Strategy Comparison

The crossover strategy plays a crucial role in determining how genetic information from the parent generation is transmitted to the offspring generation, with the optimal strategy potentially varying based on the specific characteristics of the problem at hand. In this study, we conducted comparative experiments on three representative crossover strategies: Uniform Crossover, One-Point Crossover, and Two-Point Crossover. Optimization for each strategy was carried out under fixed conditions: random seed = 42, population = 100, generation = 100. The generational convergence process for each strategy is depicted in Figure 8, which illustrates that the One-Point Crossover strategy most reliably identifies superior solutions. This suggests that the One-Point Crossover method effectively preserves the genetic information of the optimal solution during the crossover process, as it results in less disruption to the genes compared to alternative strategies.

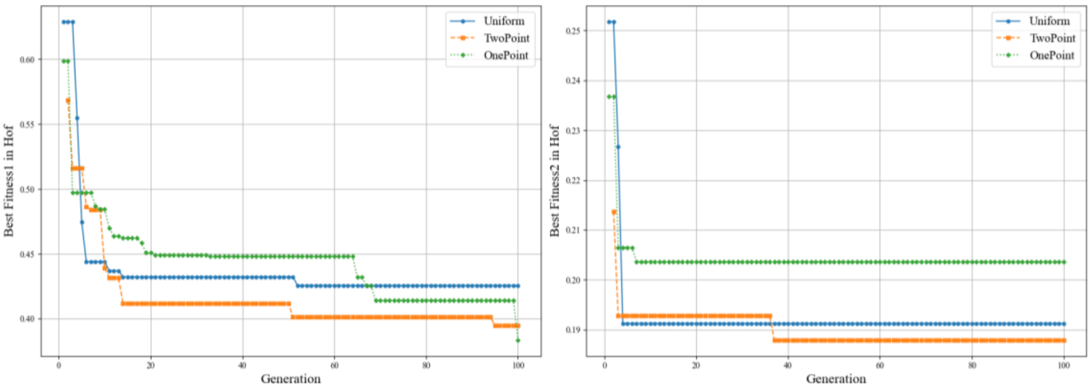


Figure 8. Comparison of Convergence Plots by Crossover Strategy

4.2.2. Parent Selection Strategy Determination

Tournament selection was employed as the method for selecting parents to generate offspring, with the tournament size serving as a critical parameter that influences the algorithm's search intensity. A large tournament size may lead to premature convergence, whereas a small size could result in reduced convergence speed. Utilizing the One-Point Crossover breeding strategy as outlined in 4.2.1, and maintaining conditions at a random seed of 42, a population of 100, and 100 generations, optimization was conducted for various tournament size values. For comparative purposes, the random selection strategy and the best selection strategy were also incorporated. Figure 9 illustrates the convergence process in relation to changes in tournament size, revealing that the most efficient search occurred when the tournament size was set to 7.

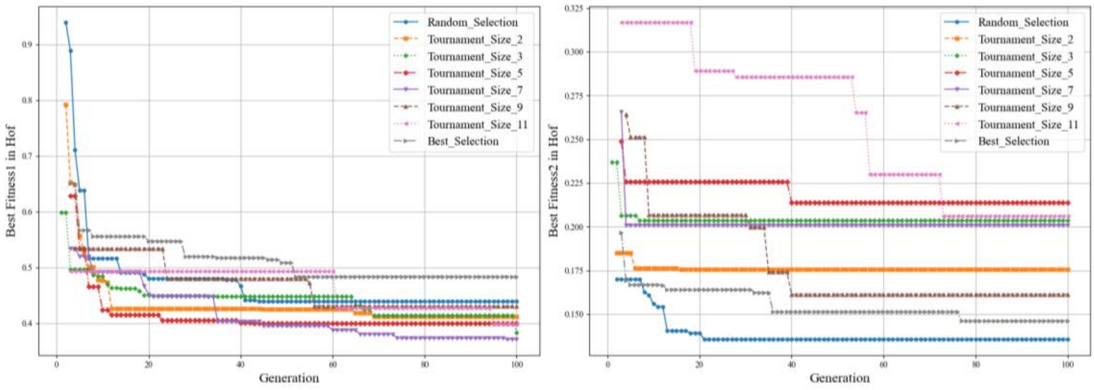


Figure 9. Comparison of Convergence Plots by Tournament Size

4.2.3. Population Size Determination

The population size is a critical parameter influencing the balance between exploration diversity and computational cost. Utilizing the optimal crossover and parent selection strategies established from prior experiments, optimization was conducted with varying population sizes. Figure 10 illustrates the detailed convergence process for each size. The optimal population size was determined to be 600, considering computational cost. This conclusion is supported by the fact that it not only achieved convergence to the lowest value, but also, beginning with a population of 600, the differences were negligible: 600 = 0.335, 700 = 0.343, 800 = 0.336, 900 = 0.347, 1,000 = 0.336.

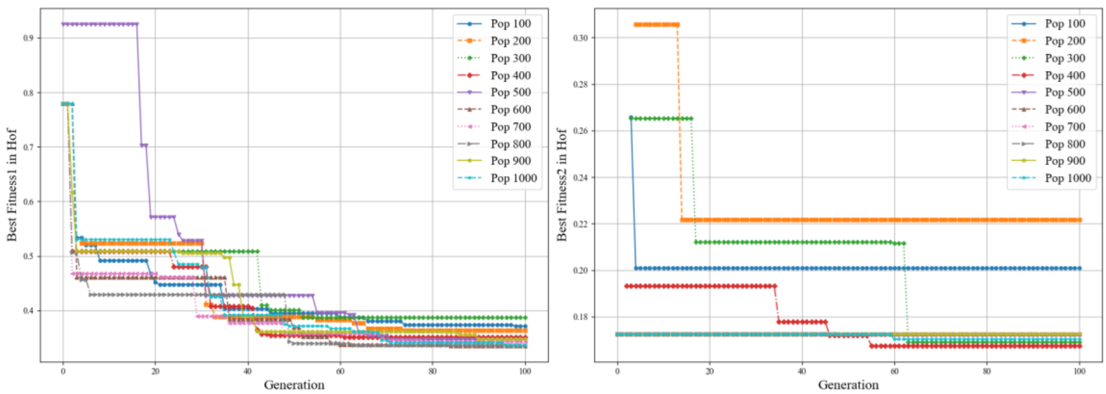


Figure 10. Comparison of Convergence Plots by Population Size

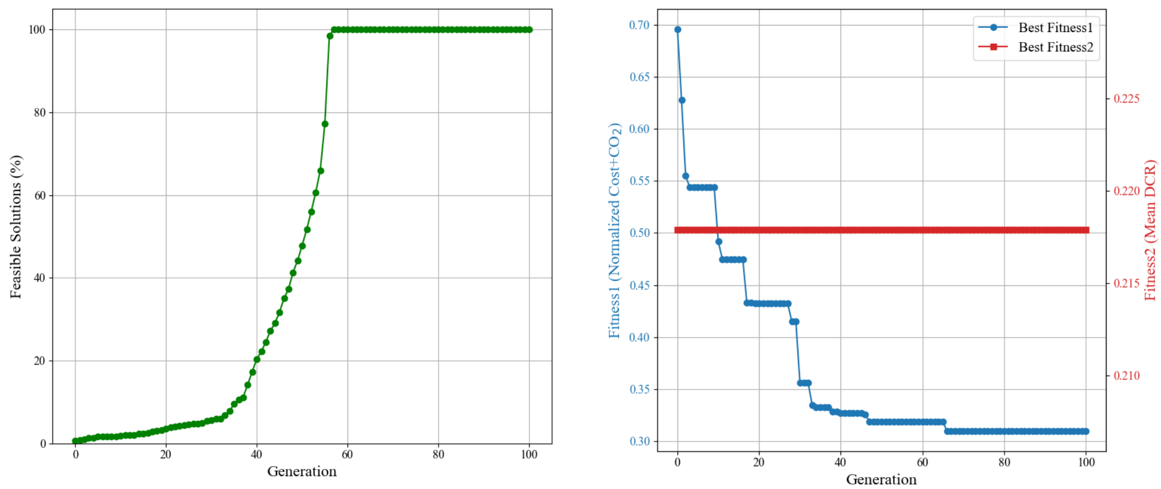
4.3. Pareto Front Analysis

The outcomes of the optimization conducted with the final selected parameters are depicted in Figure 11. As illustrated in graph (a), the algorithm initially identified a limited number of feasible solutions during the early exploration phase. However, the proportion of feasible solutions increased significantly after the 30th generation, achieving 100% before the 60th generation. This indicates that the proposed constraint-priority selection mechanism operated effectively, swiftly directing the search towards the valid design space.

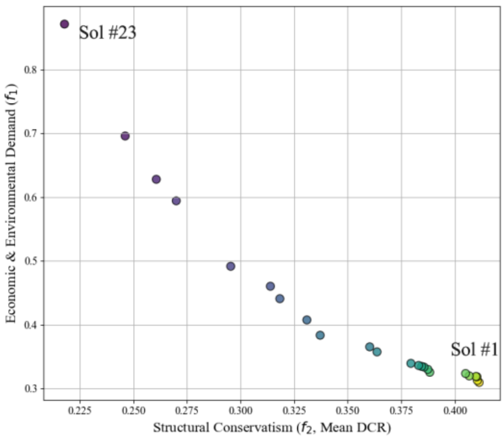
(b) The graph illustrates, for both objective functions, the optimal solution within the solution set for each generation. The first objective function (f_1) demonstrates a rapid enhancement in performance within the initial generations, after which it reaches a plateau. This suggests that the 100 generations allowed adequate time for the solutions to converge. Conversely, for the second objective function (f_2), the optimal solution from the initial population persisted through to the 100th generation, indicating that no superior average stress ratio value was identified.

(c) The graph illustrates the final results of the optimization process, presenting a total of 23 optimal solutions. The X-axis denotes structural conservatism (f_2), while the Y-axis represents economic and eco-friendly indicators (f_1). Consequently, a Pareto optimal front was successfully derived, clearly demonstrating the inverse trade-off relationship between the two objectives. For instance, the most economical solution ($f_1 \approx 0.3$), located at the lower right of the graph, exhibits an average stress ratio of approximately 0.4. In contrast, the most conservative solution ($f_2 \approx 0.2$), situated at the upper left, significantly increases the cost score to above 0.8, thereby clearly illustrating the trade-off between the two objectives.

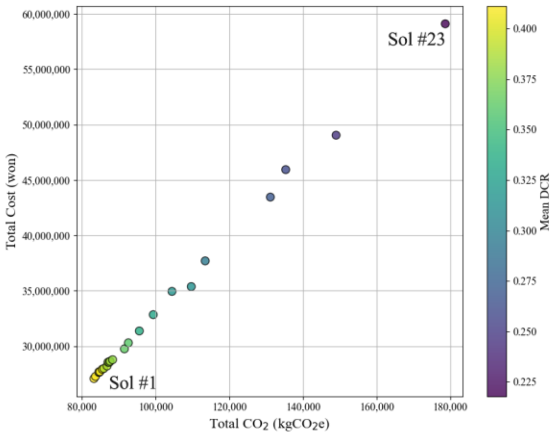
(d) The graphs illustrate the actual total construction costs and the embodied carbon associated with the optimal solutions. Furthermore, in all graphs, identical optimal solutions are represented using the same color scheme.



(a) (b)



(c)

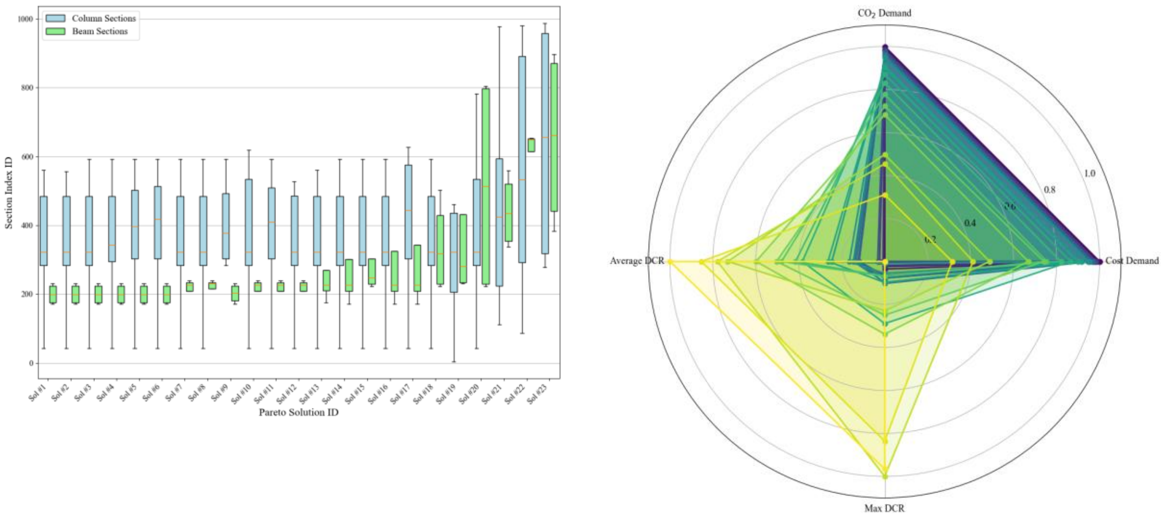


(d)

Figure 11. Optimization Process Analysis (a) Ratio of Feasible Solutions per Generation; (b) Convergence of the Hall of Fame; (c) Pareto Front in the Objective Space; (d) Pareto Front in the Solution Space (Cost vs. CO_2)

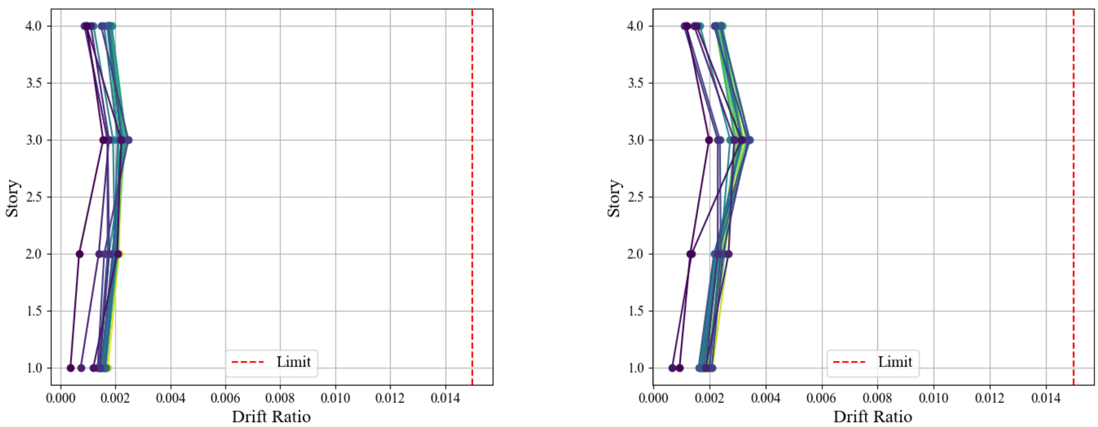
Figure 12 illustrates the distribution of design variables and performance characteristics of the optimal solutions. (a) The graph clearly indicates that the most cost-effective solution is concentrated around relatively smaller cross-sectional IDs, whereas more conservative solutions tend to select larger cross-sectional IDs. (b) The graph offers a comprehensive view of each solution's characteristics, facilitating immediate comparison: Solution #1 exhibits low values on the Cost/CO_2 Demand axis, indicating excellent cost-effectiveness, while Solution #23 shows low values on the Average Strength Ratio axis, reflecting superior conservativeness.

Figure 13 presents the outcomes of the displacement constraint analysis. The inter-story drift ratio induced by seismic loads consistently demonstrated stable performance, maintaining approximately 20–30% of the permissible limit (0.015) across all scenarios. Additionally, the top story displacement resulting from wind loads adhered to the allowable limit (0.04m) in every instance. Notably, the constraints imposed by wind loads were observed to be nearer to the limit compared to those imposed by seismic loads, suggesting that wind load design conditions exerted a more significant influence in determining the section.

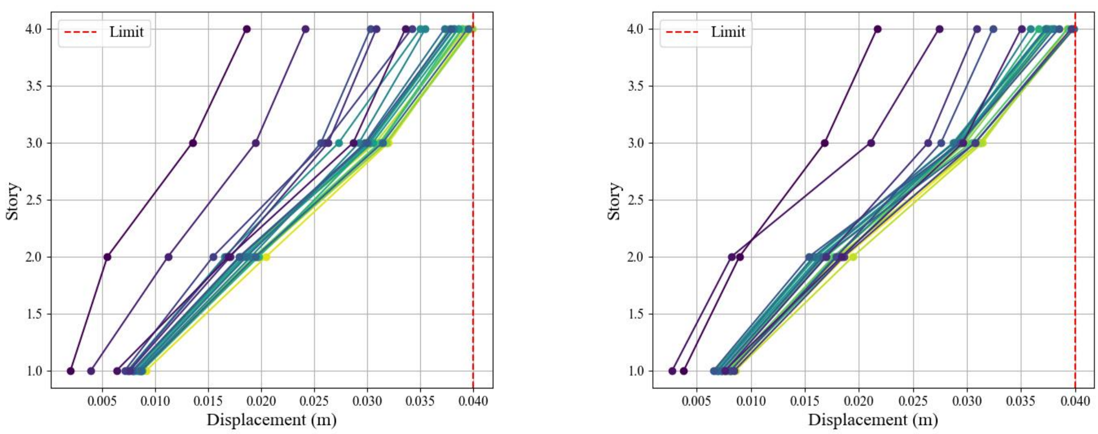


(a) (b)

Figure 12. Pareto Solutions Performance Comparison (a) Distribution of Design Variables for Each Pareto Solution; (b) Performance Comparison via Radar Chart



(a) (b)



(c) (d)

Figure 13. Displacement Checks for Pareto Solutions (a) Seismic Inter-story Drift (X-dir); (b) Seismic Inter-story Drift (Y-dir); (c) Wind Lateral Displacement (X-dir); (d) Wind Lateral Displacement (Y-dir)

4.4. Analysis of Representative Optimal Solutions

To examine the distinct characteristics of the solutions identified on the Pareto front, three representative solutions were selected for detailed comparison: Sol #1 (the most economical solution), Sol #12 (the balanced solution), and Sol #23 (the most conservative solution). These solutions are located at both extremities and the midpoint of the front. The key performance indicators for each representative solution are summarized in Table 9. The genetic information constituting each representative solution is presented in Table 10. These genetic details indicate the final section ID assigned to the member groups defined in Section 3.3.2, as well as the rotation status of the columns.

Table 8. Performance Summary of Representative Optimal Solutions

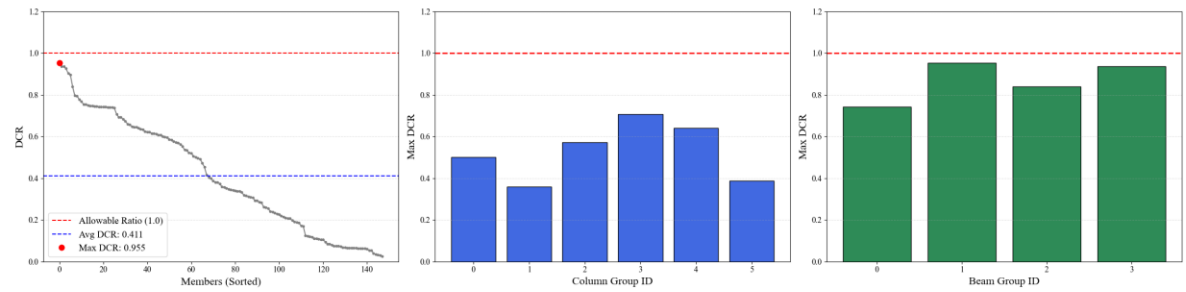
Solution ID	f_1	f_2	Total Cost (won)	Total CO_2 ($kgCO_2e$)
Sol #1	0.3096	0.4112	27,068,446	83,211
Sol #12	0.3395	0.3794	28,772,553	88,274
Sol #23	0.8715	0.2179	59,107,481	178,647

Table 9. Genetic Information of Representative Solutions

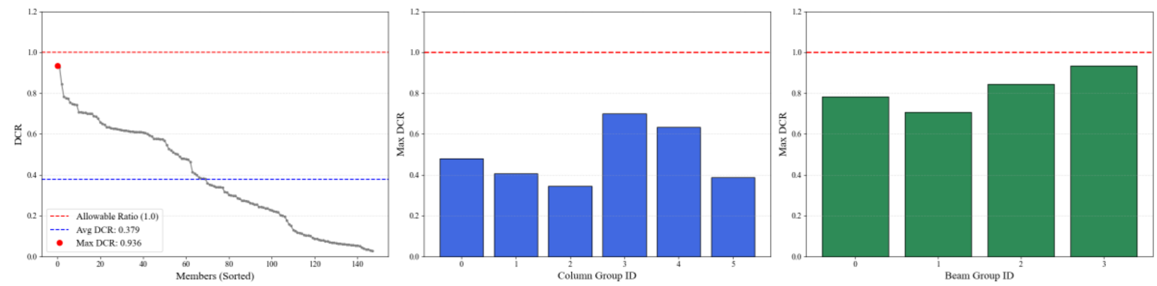
Sol #1 (Economical)		Sol #12 (Balanced)		Sol #23 (Conservative)	
Gene	Section	Gene	Section	Gene	Section
284	300 x 350	284	300 x 350	947	600 x 950
560	300 x 300	527	300 x 400	988	600 x 1200
526	300 x 400	842	300 x 350	962	550 x 950
43	300 x 300	43	300 x 300	278	300 x 550
361	300 x 300	361	300 x 300	366	450 x 600
284	300 x 350	283	300 x 300	303	350 x 450
0	-	1	-	1	-
1	-	0	-	1	-
0	-	1	-	1	-
1	-	1	-	0	-

0	-	0	-	1	-
0	-	0	-	0	-
231	300 x 350	231	300 x 350	862	600 x 1150
176	300 x 300	239	300 x 350	461	400 x 850
222	300 x 300	222	300 x 300	383	400 x 750
171	300 x 350	171	300 x 350	897	500 x 1150

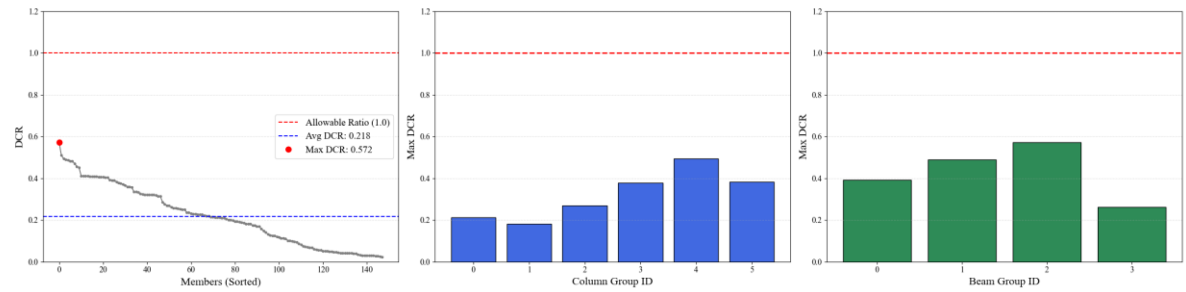
Figures 14 and 15 present a comprehensive analysis of the stress ratio and displacement for each representative solution. As illustrated in Figure 14, the economical Solution #1 demonstrates efficient material utilization, with the beam's stress ratio nearing 0.95. Conversely, the conservative Solution #23 maintains a substantial safety margin, as evidenced by the relatively low stress ratios of its members, ranging from approximately 0.2 to 0.6.



(a)

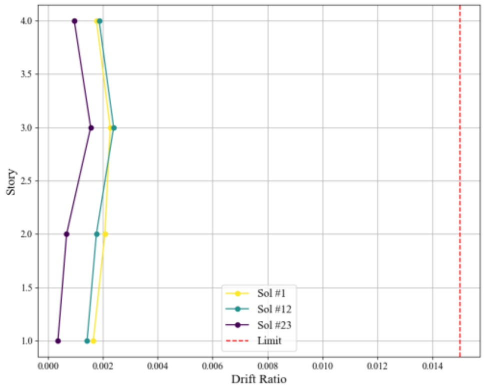


(b)

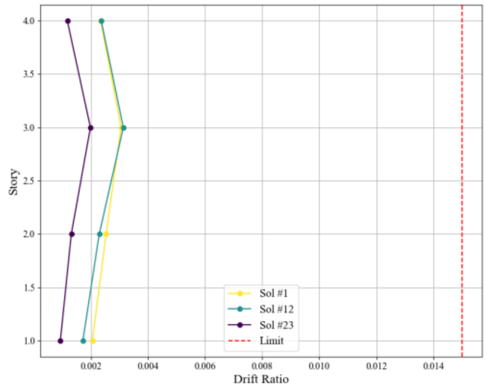


(c)

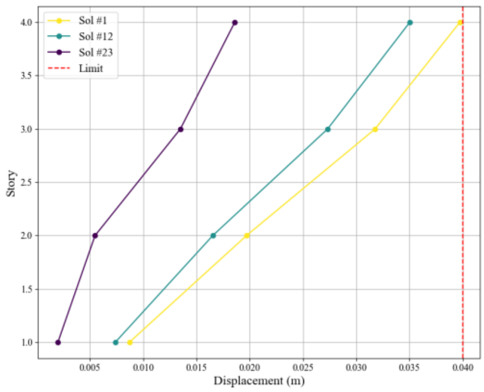
Figure 14. Detailed Stress Ratio (DCR) Analysis per Solution (a) Detailed Stress Ratio (DCR) Analysis for Sol #1; (b) Detailed Stress Ratio (DCR) Analysis for Sol #12; (c) Detailed Stress Ratio (DCR) Analysis for Sol #23



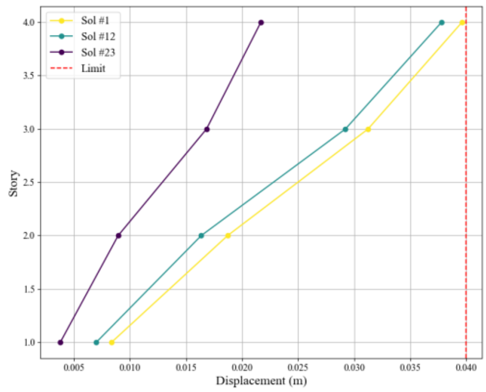
(a)



(b)



(c)



(d)

Figure 15. Displacement Checks for Representative Solutions (a) Seismic Inter-story Drift (X-dir) for Rep. Solutions; (b) Seismic Inter-story Drift (Y-dir) for Rep. Solutions; (c) Wind Lateral Displacement (X-dir) for Rep. Solutions; (d) Wind Lateral Displacement (Y-dir) for Rep. Solutions

4.5. Discussion

Through the analysis of this numerical example, the proposed optimization framework has effectively demonstrated its capability to derive a meaningful Pareto optimal front for the two conflicting objectives of economic efficiency and structural conservatism. The true value of this Pareto front lies not in its graphical representation, but in its function as a dynamic tool for evidence-based decision-making. For instance, a designer can present Figure 11c directly to a client and articulate the trade-offs in tangible terms: "As this graph shows, our most economical design (Sol #1) meets all safety codes with a cost of approximately ₩27 million. However, if we aim for a more conservative design with a 25% lower average stress ratio (from ~0.4 to ~0.3, enhancing long-term durability and user comfort), it would require an additional investment of approximately ₩15 million, leading to the design represented by Sol #18." This quantitative dialogue transforms abstract safety goals into concrete budget discussions, empowering stakeholders to make informed choices aligned with project-specific priorities.

5. Framework Performance Validation

To validate the methodological soundness and demonstrate the superiority of the proposed hybrid size-shape optimization framework, two key validation experiments were conducted. The first experiment focused on evaluating the consistency and reproducibility of the algorithm, while the second aimed to quantitatively analyze the computational efficiency of incorporating column rotation as a core design variable.

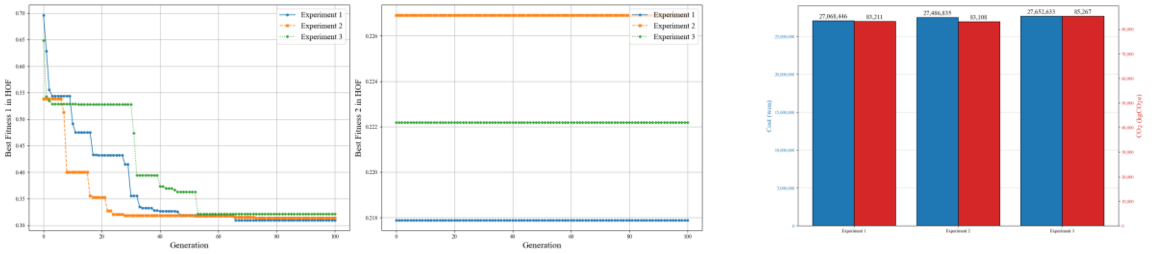
5.1. Validation of Algorithm Consistency and Reproducibility

Considering the stochastic nature of metaheuristic algorithms, it is essential to verify that they produce consistent performance across independent executions. To this end, three independent optimization trials were performed using the identical example problem and optimization parameters.

Figure 16(a) illustrates the generational convergence process for objective function 1 (the cost-carbon indicator) for all three trials. As the graph shows, while there are slight path variations in the initial exploration phase due to stochastic differences, all three trials stably converge to a narrow range between 0.31 and 0.32 after approximately 70 generations.

To assess the qualitative consistency of the final solutions, Figure 16(b) compares the actual values of the best solution (lowest cost-carbon indicator, f_1) from the Pareto front of each trial. The optimal solution from Trial 1 recorded a cost of ₩27,068,446 and 83,211$kgCO_2e$, while Trial 2 yielded ₩27,586,835 and 83,108$kgCO_2$, and Trial 3 resulted in ₩27,652,633 and 85,267$kgCO_2$. These values lie within a slight deviation of approximately 2.1% at maximum, indicating a very high level of consistency between the independent runs.

These results validate the robustness of the implemented NSGA-II algorithm. They clearly demonstrate that despite the inherent randomness of evolutionary algorithms, the proposed framework can consistently converge to high-quality, near-optimal solutions within an acceptable margin of error.



(a) (b)

Figure 16. Comparison of Three Independent Optimization Trials (a) Hall of Fame Convergence Comparison; (b) Final Solution – Actual Values Comparison

5.2. Validation of Hybrid Size-Shape Optimization Efficiency

To verify the superiority of the hybrid approach, which decouples column rotation as a separate design variable, a comparative analysis was conducted between the following two scenarios:

- Scenario A (Hybrid Approach): Utilizes the base section database (1,000 sections) and treats column rotation as a separate binary gene (total of 16 genes).
- Scenario B (Extended Database Approach): Utilizes an extended database comprising the base 1,000 column sections plus an additional 1,000 sections rotated by 90 degrees (2,000 total column sections), without a separate rotation gene (total of 10 genes).

In this comparative experiment, a unidirectional lateral load (X-direction only) was applied to maximize the effect of column rotation. This loading condition amplifies the directional stiffness requirements of the structural system, providing an optimal environment to observe the benefits of intelligent column orientation management.

Figure 17 compares the convergence performance of the two scenarios over 200 generations. The results show a distinct pattern. In the initial exploration phase (before ~25 generations), the extended database approach (Scenario B) shows temporarily superior performance due to its larger pool of initial options. However, as the algorithm begins to identify promising design regions, the hybrid approach (Scenario A) overtakes it in performance after approximately 25 generations and continues to widen the gap.

This transition stems from the core efficiency of the hybrid approach. Scenario A can quickly determine the optimal column orientations, effectively solving the 'shape optimization' problem first. It then focuses its search on 'size optimization' within a reduced search space. This dynamically reduces the dimensionality of the problem, maximizing search efficiency. Ultimately, the hybrid approach achieved an objective function value approaching 0.30, while the extended database approach plateaued around 0.33, representing a performance improvement of over 9%.

Particularly noteworthy is the convergence stability in the later generations. Whereas Scenario B exhibits a relatively unstable convergence pattern as it continues to search within a vast solution space, Scenario A stably improves its solution after the column orientations are determined. This experiment suggests that for structures with asymmetric loading conditions or irregular plans, where directional stiffness management is crucial, the hybrid approach can be a significantly more efficient and powerful optimization tool.

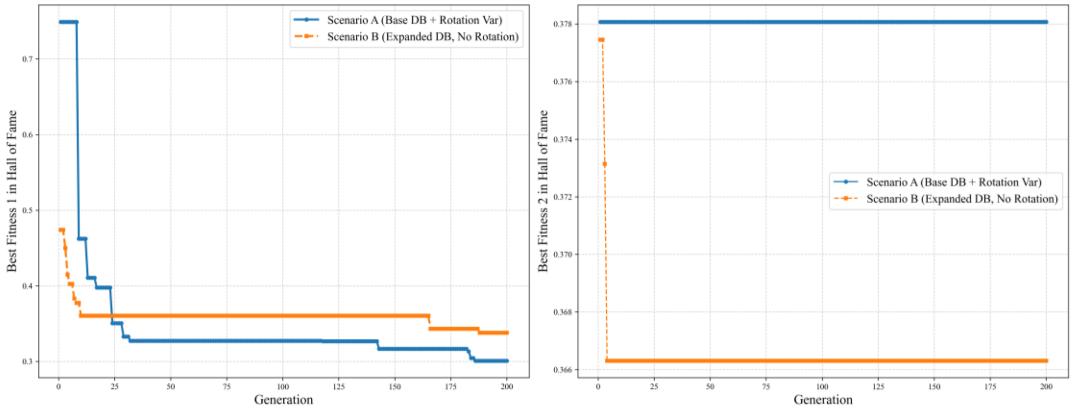


Figure 17. Convergence Performance Comparison Between Hybrid Approach and Extended Database

6. Conclusion and Future Work

This study introduced a novel hybrid size-shape optimization framework for reinforced concrete (RC) frames, defining the design variables as immediately constructible member cross-sections and the rotational status of columns. By employing the NSGA-II algorithm, the framework successfully performed a multi-objective optimization for two conflicting goals—a combined cost-carbon indicator and a structural conservatism indicator—and derived a Pareto optimal front that provides designers with a quantitative basis for informed decision-making.

Notably, this study validated the convergence stability and reproducibility of the proposed framework through independent trials. Furthermore, a comparative analysis against a traditional extended-database approach quantitatively verified the superiority of the hybrid method. The strategy of decoupling column rotation as an independent variable was shown to dynamically reduce the search space, leading to a performance improvement of over 9% and more stable convergence.

Nevertheless, this study has several limitations. First, the cross-sectional data was generated with a uniform stirrup spacing, which may not be optimal for shear resistance in all member regions. Second, costs related to rebar splicing, anchorage, and formwork were not included in the objective function. Finally, the member grouping strategy was predetermined by the user rather than being part of the optimization process.

Future research will be directed at addressing these limitations. First, the database generation process will be refined to allow for variable stirrup spacing along member lengths. The economic model will also be expanded to include costs associated with splicing, anchorage, and formwork to provide a more holistic cost assessment. Finally, an advanced optimization module will be developed to allow the algorithm to intelligently determine optimal member groupings, further expanding the search for more innovative and efficient structural systems.

Disclosure statement

Data availability

Funding

References

[1] Aga, A. A., & Adam, F. M. (2015). Design optimization of reinforced concrete frames. *Open Journal of Civil Engineering*, *05*(01), 74–83. https://doi.org/10.4236/OJCE.2015.51008

[2] Oluwole Akadiri, P., & Olaniran Fadiya, O. (2013). Empirical analysis of the determinants of environmentally sustainable practices in the UK construction industry. *Construction Innovation*, *13*(4), 352-373. https://doi.org/10.1108/CI-05-2012-0025

[3] Akin, A., & Saka, M. P. (2015). Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318-05 provisions. *Computers & Structures*, *147*, 79-95. https://doi.org/10.1016/J.COMPSTRUC.2014.10.003

[4] Aslay, S. E., Dede, T., & Kaveh, A. (2024). Integrated design optimization process for building projects. *Periodica Polytechnica Civil Engineering*, *68*(4), 1175-1183. https://doi.org/10.3311/PPCI.37113

[5] Babaei, M., & Mollayi, M. (2016). Multi-objective optimization of reinforced concrete frames using NSGA-II algorithm. *Engineering Structures and Technologies*, *8*(4), 157-164. https://doi.org/10.3846/2029882X.2016.1250230

[6] BAI, J. L., CHEN, H. M., SUN, B. H., & JIN, S. S. (2020). Seismic uniform damage-targeted design of RC frame structures. *Engineering Mechanics*, *37*(8), 179-188. https://doi.org/10.6052/J.ISSN.1000-4750.2019.10.0572

[7] Bekdaş, G. E. B. R. A. İ. L., & Nigdeli, S. M. (2014, July). Optimization of RC frame structures subjected to static loading. In *11th World Congress on Computational Mechanics* (pp. 20-25).

[8] Boscardin, J. T., Yepes, V., & Kripka, M. (2019). Optimization of reinforced concrete building frames with automated grouping of columns. *Automation in Construction*, *104*, 331-340. https://doi.org/10.1016/J.AUTCON.2019.04.024

[9] Chaudhuri, P., Barman, S., Maity, D., & Maiti, D. K. (2021). Cost effective design of RC building frame employing unified particle swarm optimization. https://doi.org/10.21203/RS.3.RS-227713/V1

[10] Chopra, A. K. (2017). Dynamics of Structures: Theory and Applications to Earthquake Engineering. Pearson.

[11] Chutani, S., & Singh, J. (2018). Use of modified hybrid PSOGSA for optimum design of RC frame. *Journal of the Chinese Institute of Engineers*, *41*(4), 342-352. https://doi.org/10.1080/02533839.2018.1473804;CTYPE:STRING:JOURNAL

[12] Coello, C. C. (2006). Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine*, *1*(1), 28-36. https://doi.org/10.1109/MCI.2006.1597059

[13] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

[14] Dehnavipour, H., Mehrabani, M., Fakhriyat, A., & Jakubczyk-Gałczyńska, A. (2019). Optimization-based design of 3D reinforced concrete structures. *Journal of soft computing in civil engineering*, *3*(3), 95-106. https://doi.org/10.22115/scce.2020.211509.1145

[15] Djedoui, N., Djafar-Henni, N., Bekdaş, G., Nigdeli, S. M., & Chebili, R. (2025). Design Optimization of Seismic Reinforced 3D Concrete Frame Structures: A Computational Framework for Cost Efficiency Using the Hybrid PSOGWO Algorithm. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 1-22. https://doi.org/10.1007/S40996-025-01915-1/TABLES/12

[16] Ehrgott, M. (2012). Vilfredo Pareto and multi-objective optimization. *Doc. math*, *8*, 447-453.

[17] Esfandiari, M. J., Urgessa, G. S., Sheikholarefin, S., & Manshadi, S. D. (2018). Optimum design of 3D reinforced concrete frames using DMPSO algorithm. *Advances in Engineering Software*, *115*, 149-160. https://doi.org/10.1016/J.ADVENGSOFT.2017.09.007

[18] Esfandiary, M. J., Sheikholarefin, S., & Bondarabadi, H. R. (2016). A combination of particle swarm optimization and multi-criterion decision-making for optimum design of reinforced concrete frames. *International journal of optimization in civil engineering*, *6*(2), 245-268.

[19] Faghirnejad, S. (2023). Performance-Based Optimization of 2D Reinforced Concrete Moment Frames through Pushover Analysis and ABC Optimization Algorithm. *arXiv preprint arXiv:2312.09450*. https://arxiv.org/pdf/2312.09450

[20] Gharehbaghi, S. (2012). Design optimization of RC frames under earthquake loads. *Iran University of Science & Technology*.

[21] Govindaraj, V., & Ramasamy, J. V. (2005). Optimum detailed design of reinforced concrete continuous beams using genetic algorithms. *Computers & structures*, *84*(1-2), 34-48. https://doi.org/10.1016/J.COMPSTRUC.2005.09.001

[22] Heydari, F., Andalib, M., Epackachi, S., & Rafiee-Dehkharghani, R. (2025). Optimized design of RC moment frames with machine learning methods. *Journal of Building Engineering*, *104*, 112222. https://doi.org/10.1016/J.JOBE.2025.112222

[23] Kaveh, A., & Ardebili, S. R. (2023). Optimal design of mixed structures under time-history loading using metaheuristic algorithm. *Periodica Polytechnica Civil Engineering*, *67*(1), 57-64. https://doi.org/10.3311/PPCI.21149

[24] Kaveh, A., Izadifard, R. A., & Mottaghi, L. (2020). Cost optimization of RC frames using automated member grouping. *International Journal of Optimization in Civil Engineering*, *10*(1), 91-100.

[25] Kaveh, A., Izadifard, R. A., & Mottaghi, L. (2020). Optimal design of planar RC frames considering CO2 emissions using ECBO, EVPS and PSO metaheuristic algorithms. *Journal of Building Engineering*, *28*, 101014.

[26] Kaveh, A., & Ardebili, S. R. (2021, December). An improved plasma generation optimization algorithm for optimal design of reinforced concrete frames under time-history loading. In *Structures* (Vol. 34, pp. 758-770). Elsevier. https://doi.org/10.1016/J.ISTRUC.2021.08.040

[27] Kaveh, A., & Ardebili, S. R. (2023, February). Optimum design of 3D reinforced concrete frames using IPGO algorithm. In *Structures* (Vol. 48, pp. 1848-1855). Elsevier. https://doi.org/10.1016/J.ISTRUC.2023.01.071

[28] Federal Emergency Management Agency (FEMA). (2012). Seismic Performance Assessment of Buildings Volume 1-Methodology. *Rep. No. FEMA P-58-1*.

[29] Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, *26*(6), 369-395. https://doi.org/10.1007/S00158-003-0368-6/METRICS

[30] Mazzoni, S., McKenna, F., Scott, M. H., & Fenves, G. L. (2006). Open system for earthquake engineering simulation (opensees) opensees command language manual. *Pacific Earthquake Engineering Research Center*, 1-465.

[31] Mergos, P. E. (2021). Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. *Journal of Building Engineering*, *44*, 102935. https://doi.org/10.1016/J.JOBE.2021.102935

[32] Mergos, P. E. (2022). Surrogate-based optimum design of 3D reinforced concrete building frames to Eurocodes. *Developments in the Built Environment*, *11*, 100079. https://doi.org/10.1016/J.DIBE.2022.100079

[33] Mergos, P. E. (2024). Structural design of reinforced concrete frames for minimum amount of concrete or embodied carbon. *Energy and Buildings*, *318*, 114505. https://doi.org/10.1016/J.ENBUILD.2024.114505

[34] Nebro, A. J., Galeano-Brajones, J., Luna, F., & Coello Coello, C. A. (2022). Is NSGA-II ready for large-scale multi-objective optimization?. *Mathematical and Computational Applications*, *27*(6), 103. https://doi.org/10.3390/MCA27060103

[35] Paya-Zaforteza, I., Yepes, V., Hospitaler, A., & Gonzalez-Vidosa, F. (2009). CO2-optimization of reinforced concrete frames by simulated annealing. *Engineering Structures*, *31*(7), 1501-1508. https://doi.org/10.1016/J.ENGSTRUCT.2009.02.034

[36] Werner, W., & Burns, J. G. (2012). Quantification and optimization of structural embodied energy and carbon. In *Structures Congress 2012* (pp. 929-940). https://doi.org/10.1061/9780784412367.083

[37] Zavala, G., Nebro, A. J., Luna, F., & Coello Coello, C. A. (2016). Structural design using multi-objective metaheuristics. Comparative study and application to a real-world problem. *Structural and Multidisciplinary Optimization*, *53*(3), 545-566. https://doi.org/10.1007/S00158-015-1291-3/FIGURES/19

[38] Zitzler, E., & Thiele, L. (2002). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, *3*(4), 257-271. https://doi.org/10.1109/4235.797969

[39] Kaveh, A., & Sabzi, O. (2011). A comparative study of two meta-heuristic algorithms for optimum design of reinforced concrete frames.

[40] McKenna, F. T. (1997). *Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing*. University of California, Berkeley.