SCIPEDIA

Cost Optimization of Reinforced Concrete Frames Using Metaheuristic Algorithms

Yasin Duysak[1,2], Sinan Melih Nigdeli[2,*] and Gebrail Bekdaş[2]

[1] Department of Civil Engineering, Faculty of Engineering, Kırklareli University, Kırklareli, Turkey

[2] Department of Civil Engineering, Faculty of Engineering, Istanbul University-Cerrahpasa, Istanbul, Turkey

INFORMATION

Keywords:
Reinforced concrete design
optimization
frame optimization
metaheuristic algorithms
structural optimization

DOI: 10.23967/j.rimni.2026.10.71897

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

In cooperation with CIMNE

Cost Optimization of Reinforced Concrete Frames Using Metaheuristic Algorithms

Yasin Duysak[1,2], **Sinan Melih Nigdeli**[2,*] **and Gebrail Bekdaş**[2]

[1]Department of Civil Engineering, Faculty of Engineering, Kırklareli University, Kırklareli, Turkey

[2]Department of Civil Engineering, Faculty of Engineering, Istanbul University-Cerrahpasa, Istanbul, Turkey

ABSTRACT

In structural engineering, the primary objective of design engineers is to ensure structural safety under applied loads according to codes and standards while achieving the most economical design. To enhance cost efficiency in complex structures such as reinforced concrete (RC) frames, computational techniques are employed. The aim is to perform a more rapid and accurate optimum cost design of RC frame systems subjected to vertical loads. This study applies five metaheuristic approaches: three metaheuristic methods and two hybrid techniques developed from them. The matrix displacement method is used to determine displacements and sectional forces of RC frame systems, with design rules following ACI 318-19 (Building Code Requirements for Structural Concrete and Commentary). The study models five different RC structures with varying dimensions and member counts, including symmetrical and asymmetrical configurations. Internal forces and displacement values of structures analyzed using the matrix displacement method in MATLAB are verified with SAP2000 structural analysis software, confirming solution validity. Among the methods applied for the optimum design, the Teaching-Learning-Based Optimization (TLBO) technique proves particularly suitable for RC frame systems. The proposed design method and TLBO algorithm offer civil engineers a rapid and efficient approach to achieving cost-optimized designs while ensuring structural safety. The developed method helps engineers efficiently solve complex design problems while achieving optimal structural solutions in terms of both cost and safety.

OPEN ACCESS

Received: 14/08/2025

Accepted: 24/10/2025

Published: 16/04/2026

DOI
10.23967/j.rimni.2026.10.71897

Keywords:
Reinforced concrete design optimization
frame optimization
metaheuristic algorithms
structural optimization

1 Introduction

In structural engineering, three significant factors in building design can be summarized as follows: structural functionality, construction cost, and the ability to safely withstand applied loads. Among these three factors, ensuring structural safety is the most significant. Reinforced concrete (RC) structures must safely provide resistance against the loads acting on them. With today's increasing

*Correspondence: Sinan Melih Nigdeli (melihnig@iuc.edu.tr).

construction costs, it has become particularly significant to achieve structurally safe buildings at reasonable costs.

Although safety and cost are often considered opposing concepts in structural system design, modern engineering approaches aim to establish an optimal balance between these two objectives. Particularly, with the advancement of material technologies, analysis methods, and artificial intelligence-assisted design, it has become possible to construct safer and more economical structures.

Today, various optimization methods are used to both ensure structural safety and design buildings at low cost. The relationship between structural safety and cost is examined in detail. Furthermore, by evaluating the methods used to optimize engineering decisions and their implications for structural design, it offers recommendations for realizing more sustainable and effective projects.

Various heuristic and mathematical algorithms are employed in civil engineering to achieve optimal solutions more efficiently. Mathematical algorithms aim to find the optimal solution by examining all possible solutions, while heuristic algorithms draw inspiration from music, genetics, physics, and natural phenomena like animal reproduction and foraging behavior to reach optimal solutions intuitively. Heuristic algorithms have a high probability of getting stuck in local solution sets. To prevent this, multiple heuristic algorithms have been combined to create and implement metaheuristic algorithms. In the study, multiple metaheuristic algorithms are utilized to achieve the optimal design of RC frame buildings subjected to vertical loads. Numerous heuristic and metaheuristic optimization algorithms, developed for different optimization purposes, are commonly used in practice.

In various countries, different design constraints exist, requiring the consideration of discrete variables for different applications. These constraints can lead to variations in solving optimization problems for reinforced concrete structures. To address this, calculations are often performed using metaheuristic algorithms, which can be easily integrated with regulatory requirements, providing flexible and efficient solutions in structural optimization.

Various optimization algorithms are commonly used in research studies including the Jaya algorithm (JA) [1] which moves toward reasonable solutions while avoiding poor ones, the teaching-learning-based optimization (TLBO) [2] inspired by classroom learning the genetic algorithm (GA) [3], the differential evolution (DE) [4], the ant colony optimization (ACO) [5], the particle swarm optimization (PSO) [6], the cuckoo search algorithm (CSA) [7], the big bang-big crunch (BB-BC) [8], the flower pollination algorithm (FPA) [9], the harmony search (HS) [10], the bat algorithm (BA) [11], the simulated annealing (SA) [12], the firefly algorithm (FA) [13], the biogeography-based optimization (BBO) [14], the whale optimization algorithm (WOA) [15] and the golden eagle optimizer (GEO) [16].

Numerous metaheuristic optimization algorithms have been developed and applied in structural engineering to improve the design efficiency of (RC) structures. These algorithms, inspired by various natural, biological, and physical processes, offer flexible and powerful tools for solving complex optimization problems. In the context of (RC) design, particularly for structural elements and systems, a wide range of studies have employed these algorithms to minimize construction cost, material usage, and environmental impact while satisfying structural performance criteria.

The literature includes numerous studies on (RC) structures, examining frame systems and individual structural elements. Particularly for columns, extensive research exists. Ref. [17] employed genetic algorithms (GA) to optimize the design and reinforcement of columns under biaxial bending moments for cost efficiency. Ref. [18] investigated slenderness effects on cost optimization by using

harmony search (HS) to design slender RC columns of varying lengths. Ref. [19] proposed Teaching-Learning-Based Optimization (TLBO) for cost-optimal design of columns with different lengths, comparing results with harmony search and bat algorithm outcomes, concluding TLBO's effectiveness for RC elements. Ref. [20] applied the metaheuristic harmony search algorithm to optimize costs in biaxially loaded RC columns. Ref. [21] utilized the bat algorithm (BA) to study cost optimization of RC columns under axial force, bending, and shear, with shorter columns showing better optimization results through BA. Ref. [22] examined environmental and construction costs of rectangular RC columns under uniaxial moment and axial load using harmony search, identifying concrete class, reinforcement area and concrete area as design variables, and demonstrating the relationship between reduced construction costs and lower environmental impact. Ref. [23] optimized RC column design under uniaxial moment and axial force using artificial bee colony algorithm (ABC) considering cross-section, reinforcement quantity and number as variables, with concrete volume, steel quantity and formwork costs as objective functions, obtaining results comparable to other studies. Ref. [24] applied the metaheuristic harmony search algorithm to optimize circular RC columns, while developing four different machine learning models using various algorithms. Ref. [25] used a genetic algorithm (GA) to optimize the design, environmental impact, and construction cost of concrete-filled composite tubular columns under buckling, bending, and axial load effects. Ref. [26] employed sequential quadratic programming (SQP) to address nonlinear problems in axially loaded RC columns, demonstrating the method's efficiency for cost optimization. Ref. [27] explored the applicability of artificial neural networks (ANN) and genetic algorithms (GA) for cost optimization in short RC columns, concluding that a hybrid ANN-GA approach was practical. Ref. [28] applied a Harmony Search (HS) algorithm to minimize the cost of RC columns under uniaxial bending, modifying the algorithm to achieve faster convergence with fewer iterations. Ref. [29] optimized the design cost of 2D RC frames under vertical and horizontal static loads using Harmony Search (HS), showing that careful column grouping significantly reduced structural costs.

The most significant cost factors in RC structures are the reinforcement and concrete used in load-bearing elements. Numerous optimization studies exist for beams, one of the most important structural elements. Ref. [30] investigated the effect of torsional moment acting on an RC beam section on the cost of the beam section using the JAYA algorithm. Ref. [31] performed an optimum cost design of the beam section under applied loads using the JAYA algorithm. Ref. [32] used the Artificial Bee Colony (ABC) algorithm to achieve economical designs for two-span and three-span continuous RC beams in terms of cross-section and reinforcement area. They modified the ABC algorithm to include a variable change percentage and demonstrated its effectiveness for optimal design of continuous RC beams. Ref. [33] used Genetic Algorithm (GA) and Simulated Annealing (SA) to study cost optimization of supported beams, showing that the proposed algorithms apply to such problems. Ref. [34] used a Genetic Algorithm (GA) to study cost optimization of supported and continuous beams, aiming to minimize construction costs by optimizing cross-section dimensions, rebar diameters, and placement. Their study showed cost improvements between 3.63% and 17.07% compared to other studies in the literature, demonstrating the algorithm's applicability to RC element optimization problems. Ref. [35] conducted optimization studies on RC load-bearing elements, using Genetic Algorithm (GA) optimization for optimal design of supported RC beams and comparing the results with those obtained by graphical methods. They concluded that the results obtained by GA were consistent with graphical methods and that the proposed algorithm could be used for RC elements. Ref. [36] studied the cost optimization of continuous RC beams under bending moment, shear force, and torsion effects using the Genetic Algorithm. They compared their design results with other studies in the literature and concluded that the proposed optimization model produced

rational, reliable, and economical solutions. Ref. [37] used the Flower Pollination Algorithm (FPA), a metaheuristic algorithm, to optimize single-span rectangular RC beams for minimum cost. Ref. [38] developed a Random Search Technique (RST) using cross-section dimensions, reinforcement amount, and concrete quality variables for optimal cost design of continuous RC beams, showing successful results. Ref. [39] used Harmony Search (HS) algorithm, a metaheuristic algorithm, to design RC beams under bending effects with different concrete strengths at minimum cost, concluding that the algorithm could be a valuable resource for preliminary design of RC elements. Ref. [40] proposed a mathematical model using Harmony Search Algorithm for cost optimization and environmental impact reduction of steel-concrete composite beams. Ref. [41] used an improved Genetic Algorithm to design RC beams under local mechanical damage and corrosion damage, concluding that improved beam design reduced repair costs despite increased initial construction costs. Ref. [42] designed prestressed concrete beams and RC beams according to ACI 318-05 code and performed cost optimization using Genetic Algorithm, showing its effectiveness for both beam types. Ref. [43] studied Eurocode 2-based design and cost optimization of RC T-beams using generalized reduced gradient technique, showing their method produced lower-cost solutions than standard office methods and was more practical than other optimization methods. Ref. [44] developed an analytical approach for rectangular RC beams with fewer design variables based on minimum cost and weight principles. Ref. [45] designed RC beams according to the Indian code IS 456:2000 and wrote analysis code using Particle Swarm Optimization (PSO) to find optimal sections, significantly reducing computation time when implemented in C++. Ref. [46] studied cost optimization and geometric layout optimization under dynamic loads for continuous RC beams using Ant Colony Optimization (ACO), demonstrating its robustness and ease of beam design. Ref. [47] performed cost optimization of rectangular RC beams considering concrete, steel, and formwork costs using Genetic Algorithm (GA). Ref. [48] conducted an optimal design study on continuous RC beams using Genetic Algorithm with only cross-sections as design variables, showing reliable and economical results. Ref. [49] performed an optimal design of singly and doubly RC beams using Artificial Neural Networks (ANN). Ref. [50] used a hybrid Firefly Algorithm (HFA) combining Simulated Annealing (SA) and standard Firefly Algorithm (FA) for optimization of RC I-beams to reduce costs and CO_2 emissions, showing better results than standard FA in both quality and computation time. Ref. [51] reviewed cost optimization studies for RC beams and performed spreadsheet-based optimization in MS Excel to demonstrate the effectiveness of nonlinear deterministic approaches.

Significant optimization studies have been conducted on RC shear walls, another crucial structural element in construction. Ref. [52] aimed to achieve both optimal seismic design and cost reduction for shear walls, columns, and beams in RC structures. Ref. [53] employed the Charged System Search Algorithm (CSSA) for seismic optimization of shear walls considering soil-structure interaction. Ref. [54] performed optimization studies on RC structures with shear walls, using wall thickness, reinforcement area, and layout as design variables. They applied optimality criteria to minimize the costs of concrete, steel, and formwork. Ref. [55] utilized an evolutionary algorithm for shear wall placement optimization in high-rise buildings, reducing structural irregularities and construction costs while streamlining the design process. Ref. [56] implemented Genetic Algorithms to optimize material costs in a 13-story RC shear wall system. Ref. [57] achieved both structural performance and cost optimization for shear walls in seismic conditions. Ref. [58] worked on optimal shear wall positioning to minimize lateral force effects according to Indian Standards.

In addition to optimization studies on structural elements, numerous optimization studies have been conducted on RC frame systems. Ref. [59] worked on cost optimization of two-dimensional RC frames. The design was performed according to ACI code and Genetic Algorithm (GA) was

used to minimize material and construction costs of the RC structure. The study demonstrates the effectiveness of genetic algorithm in the design of supported beams, uniaxial columns, and multi-story frames. Ref. [60] developed a hybrid optimization algorithm combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for design optimization of two-dimensional moment-resisting RC frames, performing nonlinear static and dynamic analyses. They showed that the hybrid GA-PSO method achieved optimal results with less computation time and material usage. Ref. [61] developed an optimization methodology for performance-based system design of RC frame systems. The optimization design method achieved structural performance design criteria under lateral loading while simultaneously optimizing construction costs for multi-story RC frame systems. Ref. [62] proposed a fully automated design method for seismic design of RC frames. The proposed optimization method provided better control of structural performance under earthquake loads and significantly reduced direct construction costs. Ref. [63] studied optimal design of 2D RC frame systems using Sequential Quadratic Programming (SQP), a nonlinear programming algorithm, achieving cost reductions up to 23%. Ref. [64] used ACI318-08 code for RC frame system design and employed Artificial Bee Colony (ABC) algorithm for optimization design. The objective function included concrete cost, reinforcement cost, and formwork cost to calculate total frame cost. They demonstrated high performance when the number of food sources was less than the bee count in the ABC algorithm. Ref. [65] used MLPSO, a hybrid algorithm combining Machine Learning (ML) and Particle Swarm Optimization (PSO), for optimization of RC frame structures under lateral seismic forces. The optimization objective was defined as minimizing material/construction costs or structural mass. Results showed MLPSO effectively produced optimal cost solutions for progressive collapse-resistant frames. Ref. [66] optimized Flower Pollination Algorithm for 2D RC frames under lateral loading according to ACI 318, demonstrating the algorithm's suitability for optimal cost design. Ref. [67] conducted an optimization study on RC frames with steel plate shear walls using Dolphin Echolocation Algorithm (DEA) and Bat Algorithm (BA), showing DEA's superior performance. Ref. [68] developed a hybrid Response Surface Method-PSO approach for element placement and cost optimization in three high-rise shear wall-frame structures, achieving 14.3% cost reduction compared to conventional methods. Ref. [69] used a hybrid multi-criteria decision-making-PSO algorithm for 3D RC frame optimization, considering element dimensions, reinforcement areas, and detailing, demonstrating efficient performance. Ref. [70] applied Flower Pollination Algorithm (FPA) to optimize design and cost of two RC frames (4- and 12-story), showing FPA's comparable or better performance than SA, GA, and PSO algorithms.

This study aims to contribute an analysis and design method to the literature that can calculate internal forces and displacements in RC space frame systems, perform design according to specified codes and regulations, and produce optimal cost solutions using metaheuristic algorithms.

One of the original aspects of this study is the verification of the MATLAB-based analysis tool (developed using the matrix displacement method) with SAP2000 software. This integration demonstrates the developed method's reliability and direct applicability to engineering practice. The study compares not just the performance of a single algorithm, but also evaluates five different optimization approaches, including hybrid algorithms. Quantitative assessment based on convergence speed, solution stability, and standard deviation values distinguishes this research from similar studies in literature.

The developed method is valuable for academic and practical applications, particularly in addressing complex structural systems' subjected to vertical loads economic and safe design challenges.

2 Methodology

The matrix displacement method offers several advantages over other classical methods (force method, moment distribution method, slope-deflection method). Due to its matrix-based computational approach, the matrix displacement method can be readily implemented numerically using MATLAB programming language, making it particularly suitable for computer-aided structural analysis applications. This method can be used for all RC space frame systems, trusses, or composite systems. Programs like SAP2000 and ETABS also use the matrix displacement method for internal force and displacement calculations. The matrix displacement method was used to determine internal forces in RC space frame systems. Structural deformations and internal forces are obtained in matrix form. The determination and organization of data required for analysis constitutes a fundamental initial step forming the basis of the optimization process. Using data received from the matrix displacement method, design and cost optimization of RC space frame systems was performed through the Jaya Algorithm, Teaching-Learning-Based Optimization Algorithm, and Flower Pollination Algorithm. In this study, a code performing the design of RC systems subjected to vertical loads using the matrix displacement method and carrying out cost optimization has been developed. For larger systems, the use of analysis software is feasible. After the systems are created in analysis programs such as SAP2000, they can be integrated into MATLAB.

2.1 Matrix Displacement Method

In RC space frame systems, external loads induce six internal force components at each end of structural members: one axial force (N_d), two shear forces (V_d), two bending moments (M_d), and one torsional moment (T_d), resulting in twelve internal force components per member. These internal forces and their positive sign convention according to the matrix displacement method are shown in Fig. 1.

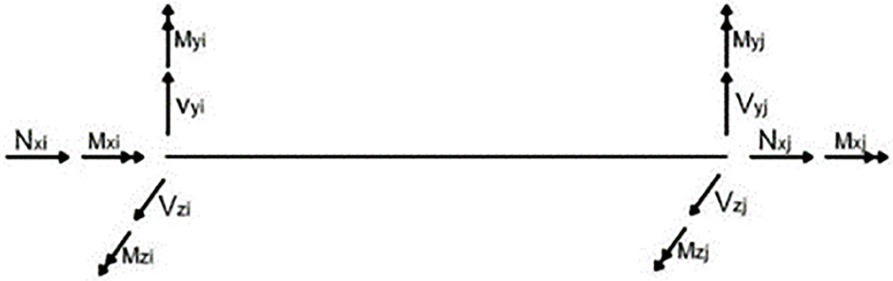

Figure 1: Internal forces in 3D elements

The analysis of RC space frame systems using the matrix displacement method is similar to the study of two-dimensional frame systems. In RC space frame systems, each member element has three linear displacements and three rotational displacements, resulting in twelve displacement components per node for each member element. The degrees of freedom for three-dimensional beam elements in space frames are shown in Fig. 2. The positive directions illustrated in the figure are valid when the member axis is connected from end (i) to end (j).

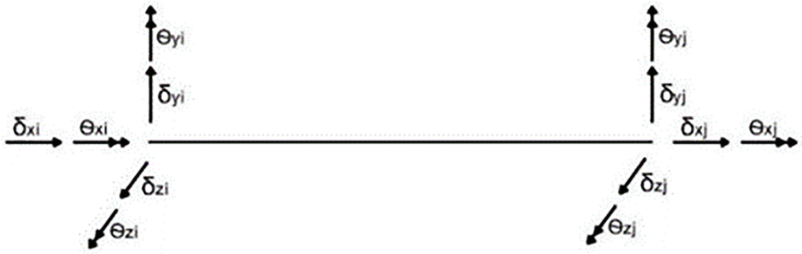

Figure 2: Degrees of freedom in 3D elements

The matrix displacement method utilizes two distinct coordinate systems. The first is the global coordinate system which encompasses the entire structure. The second is the local coordinate system defined for each structural element according to its position to determine sectional effects. These two coordinate systems are employed in structural analysis to address different requirements and play a crucial role in implementing the matrix displacement method.

Transformation matrices [T] are employed to convert between coordinate systems. In these transformation matrices, θ represents the angle between coordinate systems, while X, Y and Z denote the coordinate system axes. The transformation matrix for three-dimensional beam elements is obtained as shown in Eqs. (1) and (2).

$$T = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \end{bmatrix} \tag{1}$$

$$r = \begin{bmatrix} C_{Xx} & C_{Yx} & C_{Zx} \\ C_{Xy} & C_{Yy} & C_{Zy} \\ C_{Xz} & C_{Yz} & C_{Zz} \end{bmatrix} \tag{2}$$

$C_{Xx} = \cos\theta_{Xx}$ is defined here, where θ_{Xx}, θ_{Yx}, and θ_{Zx} angles represent the angles between the global X, Y, and Z axes and the local x-axis, respectively. In RC space frame systems, the transformation matrix elements are calculated using the equations given below.

$$C_{Xx} = (x_2 - x_1)/L \tag{3}$$

$$C_{Yx} = (y_2 - y_1)/L \tag{4}$$

$$C_{Zx} = (z_2 - z_1)/L \tag{5}$$

$$D = \sqrt{(C_{Xx} * C_{Xx} + C_{Yx} * C_{Yx})}D \tag{6}$$

$$C_{Xy} = -C_{Yx}/D \tag{7}$$

$$C_{Yy} = C_{Xx}/D \tag{8}$$

$$C_{Zy} = 0 \tag{9}$$

$$C_{Xz} = -(C_{Xx} * C_{Zx})/D \tag{10}$$

$$C_{Yz} = -(C_{Yx} * C_{Zx})/D \tag{11}$$

$$C_{Zz} = D \tag{12}$$

In RC space frame systems, the rotational and flexural stiffness factors must also be calculated to write the local stiffness matrices of the elements. Therefore, in three-dimensional systems, in addition to the cross-sectional area (A), modulus of elasticity (E), moment of inertia (I) and member length (L) of the member element, the shear modulus (G) and polar moment of inertia (J) must also be known to write the stiffness matrix $[K_L]$ in the local coordinate system. When the stiffness terms obtained in the three-dimensional member element are placed in a matrix, a 12 × 12 frame stiffness matrix with symmetric and positive elements is obtained in the local coordinate system. The local stiffness matrix is written in RC space frame systems, as shown in Fig. 3.

$$\begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{12EIz}{L^3} & 0 & 0 & 0 & \frac{6EIz}{L^3} & 0 & \frac{-12EIz}{L^3} & 0 & 0 & 0 & \frac{6EIz}{L^2} \\
 & & \frac{12EIy}{L^3} & 0 & \frac{-6EIy}{L^2} & 0 & 0 & 0 & \frac{-12EIy}{L^3} & 0 & \frac{-6EIy}{L^2} & 0 \\
 & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\
 & & & & \frac{4EIy}{L} & 0 & 0 & 0 & \frac{6EIy}{L^2} & 0 & \frac{2EIy}{L} & 0 \\
 & & & & & \frac{4EIz}{L} & 0 & \frac{-6EIz}{L^2} & 0 & 0 & 0 & \frac{2EIz}{L} \\
 & & & & & & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & \frac{12EIz}{L^3} & 0 & 0 & 0 & \frac{-6EIz}{L^2} \\
 & & & & & & & & \frac{12EIy}{L^3} & 0 & \frac{6EIy}{L^2} & 0 \\
 & & & & & & & & & \frac{GJ}{L} & 0 & 0 \\
 & & & & & & & & & & \frac{4EIy}{L} & 0 \\
sym & & & & & & & & & & & \frac{4EIz}{L}
\end{bmatrix}$$

Figure 3: Local stiffness matrix

In frame structural systems, the stiffness matrix for each member element in the global coordinate system is calculated using Eq. (13). After determining the global stiffness matrix of each member element, they are superimposed, and by considering the boundary conditions, the relevant rows and columns are eliminated from the matrix to form the system's global stiffness matrix $[K_{GS}]$.

$$[K_G] = [T]^T * \lceil K_L \rceil * [T] \tag{13}$$

The forces acting on the structure are expressed in vector form. The load vector must have dimensions of 3n × 1 for 2D systems and 6n × 1 for 3D systems, where (n) represents the number of member elements. Using Eq. (14), the frame system's global displacement vector $[\Delta_G]$ is calculated. In the equation, $[P_G]$ denotes the external loads acting on the system in the global coordinate system.

$$[\Delta_G] = [K_{GS}]^{-1} * [P_G] \tag{14}$$

To determine the internal forces of member elements in the frame system, a backward solution is performed using Eq. (15). The local displacement vectors $[\Delta_L]$ are calculated by multiplying the transformation matrix by the global displacement vector.

$$[\Delta_L] = [T] * [\Delta_G] \tag{15}$$

The internal forces of member elements are obtained by multiplying the local stiffness matrix of each member with its local displacement vector using Eq. (16). In Eq. (16), $[K_L]$ represents the local stiffness matrix of the member element, $[\Delta_L]$ denotes the displacement vector occurring in the member element, and $[P_L]$ indicates the internal force vector developed in the element.

$$[P_L] = [K_L] * [\Delta_L] \tag{16}$$

2.2 Jaya Algorithm

The Jaya algorithm is an optimization method that simulates collective behavior in populations. It moves away from poor solutions while approaching better ones [1]. This principle gives the Jaya algorithm effective convergence capabilities. The Jaya algorithm can generally be examined in five steps, as described below.

i. Definition of population size, iteration count, design variables, and stopping criteria

This stage determines fundamental parameters such as population size (number of candidate solutions) and termination criteria (maximum iteration count). Design variables are specified and randomly generated within their lower and upper bounds. Solution vectors are created according to the defined population size. Analysis and design processes are executed for each solution vector and objective function values are computed. These calculated values are stored in the objective function vector.

ii. Identification and recording of the current best and worst results

The best and worst solution values from the objective functions calculated in the first step are recorded.

iii. Generation of new results based on the best and worst outcomes

In this stage, new design variables are randomly regenerated by considering both the design variables yielding the best and the worst results. The newly generated values are checked to verify they remain within the lower and upper limits. Using Eq. (17), the new design variables produced in each iteration are calculated.

$$x_i^{j,t+1} = x_i^{j,t} + r_1 \left(x_i^* - \left|x_i^{j,t}\right|\right) - r_2 \left(x_i^w - \left|x_i^{j,t}\right|\right) \mathrm{i} = 1, 2, \ldots, \mathrm{n}; \mathrm{j} = 1, 2, \ldots, \mathrm{p}; \mathrm{t} = 1, 2, \ldots, \mathrm{tmax} \tag{17}$$

iv. Update of the objective function vectors

If the newly generated objective function vectors are superior to the initially produced ones, the old vectors are deleted and replaced with the newly generated solution vectors. This indicates that the solution vectors have been revised to achieve better results. However, if the initially generated objective function vectors remain more suitable than the newly generated ones the original objective function vectors are retained unchanged.

v. The algorithm verifies whether the stopping criteria are satisfied.

The above-described operations are repeated from Step 2 until reaching the maximum iteration count specified by the user. The user-defined maximum iteration number serves as the termination condition. The obtained results are the optimal solution upon satisfying the stopping criterion.

2.3 Flower Pollination Algorithm

A global optimization search is performed in the flower pollination algorithm's initial stage. The long-distance pollen transfers by abiotic pollinators such as flies, bees, and insects guarantee optimal reproduction. The algorithm models these long-distance insect movements during pollination using

Lévy distribution [9]. The global optimum is shown in Eq. (18).

$$x_i^{j,t+1} = x_i^{j,t} + L\left(x_i^{j,t} - g_i^*\right) \mathrm{i} = 1,2,\ldots,\mathrm{n};\mathrm{j} = 1,2,\ldots,\mathrm{p};\mathrm{t} = 1,2,\ldots,\mathrm{tmax} \tag{18}$$

In the above equation, L represents the Lévy distribution describing the random flight patterns of pollinators like flies and insects. At the same time, g_i^* denotes the design variable yielding the best solution in the current population solution matrix. The mathematical expression of the Lévy distribution is given in Eq. (19).

$$L = \frac{1}{2\pi} * \left(r^{-1.5}\right) * e^{-\frac{1}{2r}} \tag{19}$$

The equation above, the variable r represents randomly generated numbers between 0 and 1. The local optimization process mimics the abiotic characteristics of flowering plants, where abiotic pollination refers to self-fertilization without pollinators, as formulated in Eq. (20).

$$x_i^{j,t+1} = x_i^{j,t} + \varepsilon\left(x_i^{a,t} - x_i^{b,t}\right) \mathrm{i} = 1,2,\ldots,\mathrm{n};\mathrm{j} = 1,2,\ldots,\mathrm{p};\mathrm{t} = 1,2,\ldots,\mathrm{tmax} \tag{20}$$

In the above equation, ε represents a randomly selected number between 0 and 1. Two random individuals ($x_i^{a,t}$ and $x_i^{b,t}$) are chosen from the current population matrix to represent pollination within the same plant species. If solutions obtained after these two stages yield better results with newly calculated values, they are updated accordingly. The optimization continues until the specified iteration count is reached or the defined design criterion is satisfied.

2.4 Teaching-Learning-Based Optimization

Another metaheuristic algorithm developed [2] for solving complex optimization problems is the Teaching-Learning-Based Optimization (TLBO) algorithm. Inspired by knowledge acquisition in classroom settings, TLBO consists of two phases: the teacher phase (knowledge transfer from instructor to students) and the learner phase (peer interaction among students). The teacher phase represents global optimization search, while the learner phase corresponds to local optimization search. The mathematical formulation of the teaching phase is presented in Eq. (21).

The mathematical formulation of the algorithm's teaching phase is shown in Eq. (21).

$$x_i^{j,t+1} = x_i^{j,t} + \mathrm{rand}\left(g_i^x - TF_{xi}^{ave}\right) \mathrm{i} = 1,2,\ldots,\mathrm{n};\mathrm{j} = 1,2,\ldots,\mathrm{p};\mathrm{t} = 1,2,\ldots,\mathrm{tmax} \tag{21}$$

In the first phase of the algorithm, the best solution g_i^* in the solution matrix is used since the teacher possesses the optimal knowledge. TF (teaching factor) is a parameter not specified by the user. The teaching factor in the formula represents a randomly selected number between 1 and 2. Unlike other two-phase algorithms, the teaching-learning-based optimization algorithm does not require choosing between global and local searches. Both phases are performed in each iteration, and the solution is updated whenever either phase yields improved results.

The learning phase represents the local optimization component of this algorithm. Eq. (22) shows the formula for achieving correct knowledge through student interactions.

$$\boldsymbol{x}_i^{j,t+1} = \begin{cases} \mathbf{x_i^{j,t}} + \mathbf{rand}\,(\mathbf{1})\left(\mathbf{x_i^{a,t}}\mathbf{x_i^{b,t}}\right) \mathbf{if\ f}\left(\mathbf{x_i^{a,t}}\right) < \mathbf{x_i^{b,t}} \\ \mathbf{x_i^{j,t}} + \mathbf{rand}\,(\mathbf{1})\left(\mathbf{x_i^{a,t}}\mathbf{x_i^{b,t}}\right) \mathbf{if\ f}\left(\mathbf{x_i^{a,t}}\right) > \mathbf{x_i^{b,t}} \end{cases} \tag{22}$$

Similar to the flower pollination algorithm, the local optimization search selects two random solution individuals ($\mathbf{x_i^{a,t}}$ and $\mathbf{x_i^{b,t}}$) from the current solution matrix. These represent students who have

acquired specific knowledge after the teaching phase and now interact to reach the correct knowledge and learning levels. After both phases, the obtained solution is compared with the previous one, and the better solution is retained.

The sequential combination of teaching and learning phases enables global and local optimum solution searches. At each step, the algorithm aims to approach more optimal solutions in the search space to obtain high-quality results. The optimization process continues until the specified iteration count is reached or the defined design criterion is satisfied.

2.5 Reinforced Concrete Member Design Process and Calculation Methods

The design of RC frame systems is a significant and equally complex problem in the field of structural engineering. Many countries derive their regulations or adopt and revise other regulations to design frame systems. The design and construction rules in the codes used to design RC structures are established to ensure that the structures safely carry the loads acting on them. In the design of RC structures, the ACI-318 (Building Code Requirements for Structural Concrete and Commentary) regulation is widely referenced in international projects [71]. Since many countries directly adopt the ACI-318 code, this standard becomes a frequently used guide in RC design. The ACI-318-19 code is considered when determining design constraints and construction rules. In the design of RC frame systems, issues such as durability, safety, and structural integrity are addressed according to ACI-318-19. This code is a reliable reference in projects as it contains standards for many details, from the sizing of RC elements to material properties. For the design of RC space frame systems, the coordinates of the column and beam connection points, the number and coordinates of the nodal points of the structure, and the boundary conditions of the elements are first determined. In the RC frame system, the cross-sectional dimensions are first defined. Then the reinforcement is designed according to the internal forces resulting from the static analysis of the system. The cross-sectional width (b_w) and cross-sectional height (h) of RC elements are design variables. The design variables remain within specific values and are generated randomly. The minimum and maximum values of the element cross-sections are shown in Eqs. (23) and (24).

$$300^{mm} \leq b_w \leq 600^{mm} \tag{23}$$

$$300^{mm} \leq h \leq 600^{mm} \tag{24}$$

For the static analysis, the loads acting on the structure are defined. In the load definition, G designates the dead load (D) while Q represents the live load (L). When the applied loads are relatively small, the cross-sectional dimensions of columns and beams can be designed to meet the minimum requirements specified by design codes. Conversely, higher load magnitudes—particularly on columns—necessitate maximum permissible cross-sectional dimensions for structural members. To systematically investigate the variation in member cross-sections using the adopted design methodology, uniformly distributed loads of DL = 100 kN/m and LL = 50 kN/m are applied to all beam elements. The load combination used for static analysis is given in Eq. (25).

$$F_d = 1.4G + 1.6Q \tag{25}$$

The design parameters for static analysis have been established. The design constants used in frame systems are expressed as: concrete unit weight (γ_c), reinforcement unit weight (γ_s), modulus of elasticity of reinforcement (E_s), shear modulus (G), clear cover thickness (d′), concrete compressive strength ($f_c^{'}$), reinforcement yield strength (f_γ), cost per cubic meter of concrete (C_c), cost per ton of reinforcement (C_s), and stirrup diameter (Ø).

The internal forces obtained from the matrix displacement method enable the design of columns and beams in RC frame systems.

As shown in Fig. 4, a compression zone develops in the upper portion of the RC member above the neutral axis, and the resultant force is designated as F_c. Below the neutral axis, a tension zone exists, with its corresponding force denoted as F_s. When loads are applied to the RC beam, the concrete in the upper region resists compressive forces, while the reinforcement in the lower region carries tensile forces. These opposing forces in the two zones maintain equilibrium in the structural member and enable it to sustain applied loads.

$$F_c = a * f_c^{'} * b * a \tag{26}$$

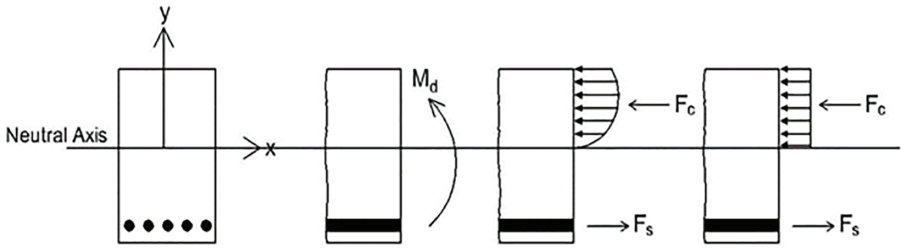

Figure 4: Pressure zone load distribution [37]

In Eq. (26), the concrete compressive strength is multiplied by the α coefficient (0.85). The parameter a in Eq. (26) represents the depth of the equivalent rectangular pressure block.

$$\mathrm{a} = \beta_1 * c \tag{27}$$

In Eq. (27), the variable c represents the neutral axis depth within the compression zone. The ACI 318 standard defines β_1 as the equivalent rectangular pressure block parameter, as given in Eq. (28).

$$\beta_1 = 0.8517 \text{ MPa} < f_c^{'} < 28 \text{ MPa}$$
$$\beta_1 = 0.85 - 0.0071428\left(f_c^{'} - 28\right) f_c^{'} > 28 \text{ MPa} \tag{28}$$

The tensile force in the reinforcement occurring in the tension zone is shown in Eq. (29).

$$F_s = A_s * f_y \tag{29}$$

The tensile reinforcement area (A_s) shown in Eq. (29) represents the cross-sectional area of tension reinforcement in concrete. When the concrete's compressive strain (ε_c) reaches its ultimate value of 0.003, the tensile steel strain (ε_s) simultaneously reaches the yield strain (ε_{sy}). This condition is called a balanced failure condition. According to ACI-318 code, beam designs must consider underbalanced failure in calculations. The underbalanced failure calculations are performed based on the balanced reinforcement ratio.

Considering the strain condition and internal forces shown in Fig. 5, the balanced reinforcement ratio ρ_b that determines the balanced condition can be calculated according to Eq. (30) when the material properties are known.

$$\rho_b = 0.85 * \beta_1 \frac{f_c^{'}}{f_y}\left(\frac{600}{600 + f_y}\right) \tag{30}$$

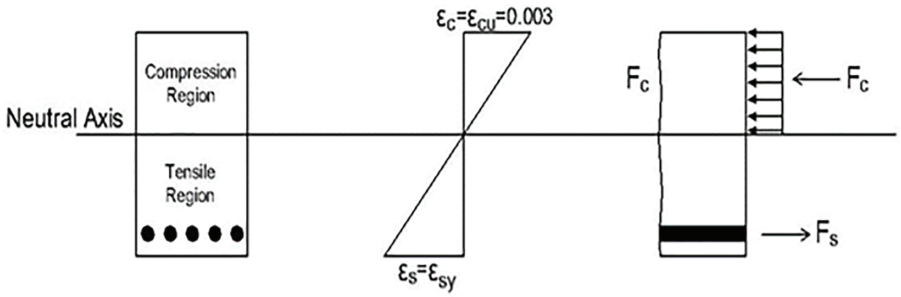

Figure 5: Beam cross-section and deformation state [37]

According to ACI-318 code, to ensure ductile behavior in beams, the maximum reinforcement ratio $\rho_{\max}$ shall not exceed $0.75\rho_b$ and shall not be less than $0.025\rho_b$. These limiting values for the reinforcement ratio have been established to obtain beam sections with ductile behavior.

According to ACI-318 code, the minimum longitudinal reinforcement ratio for beams is calculated based on Eqs. (31) and (32).

$$A_{s,min} \geq \frac{\sqrt{f'_c}}{4f_y} bd \tag{31}$$

$$A_{s,min} \geq \frac{1.4}{f_y} bd \tag{32}$$

The calculated reinforcement ratio shall be taken as the minimum longitudinal reinforcement ratio if it is lower than the minimum ratios given in Eqs. (31) and (32). If the longitudinal reinforcement ratio exceeds the maximum reinforcement ratio, the section dimensions shall be modified and calculations repeated. The stirrup design shall be performed after determining the required longitudinal reinforcement for the beam. Stirrup design shall comply with the rules and limitations specified by the relevant code.

The maximum shear strength provided by concrete

$$V_c = 0.17\sqrt{f'_c} bd \tag{33}$$

The maximum shear strength provided by reinforcement

$$V_s = 0.67\sqrt{f'_c} bd \tag{34}$$

Minimum stirrup area

$$A_{v,min} = 0.35\frac{b_w s}{f_{yt}} \tag{35}$$

The variable s in Eq. (36) represents the spacing between stirrup reinforcements. The spacing between shear reinforcements

$$s \leq min\{d/4, 8\varnothing_{min}, 150_{mm}\} \tag{36}$$

must satisfy the conditions.

The support, span, and shear reinforcement quantities for each beam were calculated in tons. Then, the total reinforcement quantity for each beam was recorded in tons, and the total cost of each beam was calculated. The total cost of beam elements is shown in Eq. (37).

$$B_{cost} = A_c * L * C_c + B_s * C_s \tag{37}$$

In the above equation, B_{cost} represents the total cost of the beam element, A_c denotes the cross-sectional area of the element, L is the length of the element, and B_s indicates the total amount of reinforcement in the beam element calculated in tons.

Similar to beams, column cross-sections are generated randomly. As with beams, the maximum and minimum column dimensions are specified in Eqs. (23) and (24). Following the same approach as for beams, column dimensions are randomly generated at 5 cm intervals to facilitate construction.

According to ACI-318-19, the minimum number and diameter of longitudinal reinforcement must not be less than either 4ϕ16 or 6ϕ14. Due to symmetrical reinforcement requirements, the minimum longitudinal reinforcement for columns is selected as 4ϕ16 in diameter and quantity.

The ACI-318 code recommends using the moment magnification method for columns carrying axial compression and bending moments. The axial force and moment values obtained from static analysis are not used directly in structural design. To better reflect the actual behavior of the structure, we must consider the second-order effects of the obtained normal force and moment values. Second-order moments are the structure that generates additional moments while deforming under applied loads. Our frame system is designed as a laterally unbraced system since there are no large shear walls in our frame system and no restraining system limiting the lateral movement of our structure.

The moment magnification method first requires determining the column effective length. The column effective length (l_k) is obtained by multiplying the column free length (l_n) by a k coefficient, as shown in Eq. (38).

$$\begin{aligned} &a_m < 2 \text{ ise}, k = \frac{20 - a_m}{20}\sqrt{1 + a_m} \\ &a_m \geq 2 \text{ ise}, k = 0.9\sqrt{1 + a_m} \end{aligned} \tag{38}$$

In the moment magnification method, for columns with one pinned end in laterally unbraced frames, k is calculated as $k = 2 + 0.3\alpha_2$. The α_m parameter used in Eq. (38) is calculated according to Eqs. (39) and (40) shown below.

$$a_{1,2} = \frac{\sum \left(\frac{\mathrm{EI}}{l}\right)_{kolon}}{\sum \left(\frac{EI}{l}\right)_{kiris}} \tag{39}$$

$$a_m = 0.5 * (a_1 + a_2) \tag{40}$$

For the $\alpha_{1,2}$ ratios in Eq. (39), the beams acting in the column's moment direction must be considered. In some cases, the slenderness effect may be neglected. For laterally unbraced columns, if $(l_k/i) \leq 34 - 12(M_1/M_2) \leq 40$, the moment magnification method is not used and moment magnification is neglected. For slender columns, the column buckling load is calculated using the Euler equation shown in Eq. (41) to determine the moment magnification factor.

$$N_k = \frac{\pi^2 \mathrm{EI}}{l_k^2} \tag{41}$$

The moment modification factor in ACI-318 is calculated as shown in Eq. (42).

$$C_m = 0.6 - 0.4 * \frac{M_1}{M_2} \tag{42}$$

In Eq. (42), M_1 and M_2 represent the first-order moments calculated from static analysis. In the equation, M_1 is always taken as the smaller absolute value of the column end moments. Finally, the moment magnification coefficient is calculated according to the equation in Eq. (43).

$$\delta = \frac{C_m}{\frac{1 - N_d}{0.75N_k}} \geq 1.0 \tag{43}$$

In the above equation, δ represents the moment magnification coefficient. If the calculated value is less than 1, the moment magnification coefficient should be 1. The new calculated column end moment is determined as shown in Eq. (44). Furthermore, according to ACI-318 provisions, the column end moment value must not be less than the value specified in Eq. (45). When the moment magnification coefficient δ exceeds 1.4 the column cross-sections must be enlarged.

$$M_c = \delta * M_2 \tag{44}$$

$$M_{min} = N_d * (15 + 0.03h) \tag{45}$$

In Eq. (45), N_d represents the axial load acting on the column, while h denotes the cross-sectional dimension of the column in the direction being analyzed. In the equation, the values 15 and h are in millimeters.

Eq. (46) shows the minimum and maximum longitudinal reinforcement ratios applicable to RC columns. The randomly generated column dimensions are multiplied by the reinforcement ratios and checked against the interaction diagram to verify that they fall within the usable region.

$$\rho_{min} = 0.01, \rho_{max} = 0.06 \tag{46}$$

The shear reinforcement calculations are performed after calculating and recording the longitudinal reinforcement quantities in columns. The shear reinforcement design for columns follows the same procedure as for beams, using Eqs. (33)–(36). The calculated longitudinal and shear reinforcement quantities for columns are recorded in tons and the total cost for each column is determined. The total cost of column elements is shown in Eq. (47).

$$C_{cost} = A_c * L * C_c + C_{ls} * C_s \tag{47}$$

In the above equation, C_{cost} represents the total cost of the column element, A_c denotes the cross-sectional area of the element, L indicates the element length, C_{ls} is the total reinforcement quantity in the column element calculated in tons, C_c is the cost of concrete per m^3 and C_s is the cost of reinforcement per ton. After calculating the costs of columns and beams, the total cost of the structure is determined as shown in Eq. (48).

$$TM_{\$} = \sum_1^n (B_{cost})_i + \sum_1^n (C_{cost})_i \tag{48}$$

Moreover, Eq. (48) serves as the objective function of the study. Throughout the optimization process the solution yielding the best objective function value is recorded and continues until the specified stopping criteria are met.

3 Numerical Example

This section calculates the optimum cost of RC space frame systems subjected to vertical loads using metaheuristic algorithms. Within this scope, the Flower Pollination Algorithm (FPA), Teaching-Learning Based Optimization Algorithm (TLBO), and Jaya Algorithm (JA) have been employed. The standalone performance of the Jaya Algorithm (JA) appears weaker compared to other algorithms; however, its structural simplicity and parameter-free design make it a strong candidate for hybridization. With no requirement for user-defined parameters, JA allows more straightforward implementation and integration with other optimization methods. The principle of moving towards the best solution while avoiding the worst provides a flexible yet straightforward foundation for hybrid models. By incorporating the strengths of the Flower Pollination Algorithm (FPA) and the Teaching–Learning-Based Optimization (TLBO), two hybrids, JA/FPA and JA/TLBO, emerge as enhanced variants. These hybrids reduce the likelihood of entrapment in local optima while improving both solution quality and search efficiency. For these reasons, JA serves as a suitable base algorithm for hybridization. A total of five different algorithms including the hybrid ones, are utilized.

To identify the algorithm providing the most suitable solution, two structures a two-story 42-element structure and a 58-element structure, have been optimized and their results examined. Table 1 presents the design constants and variables used for the static analysis of RC structures. All static analyses conducted have employed concrete class C30 and reinforcing steel with a yield strength of 420 MPa.

Table 1: Design constants and design variables

Explanation	Symbol	Unit	Values
Section width	b_w	mm	300–600 mm
Section height	h	mm	300–600 mm
Concrete compressive strength	f_c	MPa	30 MPa
Steel yield strength	f_y	MPa	420 MPa
Concrete modulus of elasticity	E	MPa	27,800 MPa
Concrete unit weight	γ_c	t/m^3	2.5 t/m^3
Steel unit weight	γ_s	t/m^3	7.86 t/m^3
Clear cover	d	mm	40 mm
Stirrup diameter	$Ø$	mm	8–14 mm
Concrete unit cost (per m^3)	$	USD	78
Reinforcement unit cost (per ton)	$	USD	676

The cross-sectional dimensions of RC structural elements—width (b_w) and height (h)—have been limited to a minimum of 300 mm and a maximum of 600 mm. For the optimum cost design, the cross-sectional dimensions of structural elements have been treated as discrete variables, implemented with 50 mm increments within the defined range of 300 to 600 mm. Another discrete variable used in the cost design has been the stirrup diameter, with structural design performed using stirrup diameters ranging between 8 and 14 mm. The calculated reinforcement quantity for columns and beams determines the minimum reinforcement area required in the section. Since the diameter and quantity of the longitudinal reinforcements are not specified in detail, no constraints are imposed regarding the minimum and maximum spacing between the reinforcements. In the analyses, a personal computer

with an Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz (2.59 GHz) processor and 16.0 GB RAM was used.

3.1 Example 1

This section examines optimizing a two-story RC frame system with two spans in both directions. The structure contains 18 columns and 24 beams, totaling 42 structural members. The design features 4 m span lengths and 3 m story heights. Fig. 6 displays the frame system elevation, loading conditions, and cross-section along the x-axis. Fig. 6 represents a simple two-story reinforced concrete structure under vertical load only. Each story of the structure contains 9 columns and 12 beams. Additionally, Fig. 6 shows the structure cross-section and the vertical loads acting on the structure. The structure is subjected to a 100 kN/m dead load and 50 kN/m live load. The 1.4G + 1.6Q load combination has only been applied for vertical loads.

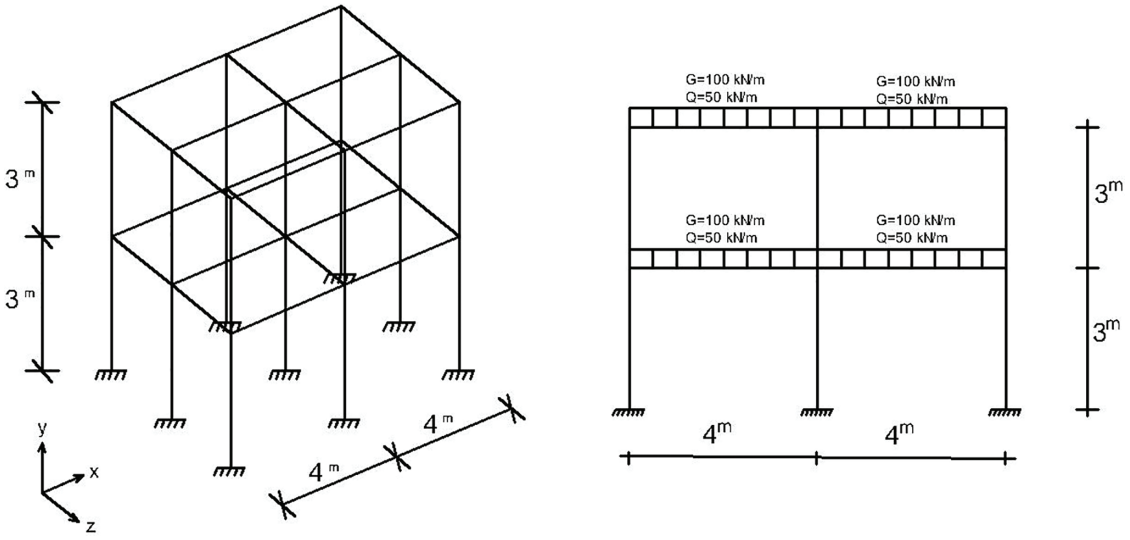

Figure 6: Load distribution and cross-section view of the 2-storey 2-span structure

Fig. 7 presents the floor plan, showing columns and beams of the RC frame system, as illustrated in Fig. 6. In Fig. 7, labels starting with 'S' (e.g., S1, S2, S3, ...) represent the reinforced concrete columns, while labels starting with 'K' (e.g., K1, K2, K3, ...) represent the beams. All columns maintain constant cross-sectional dimensions throughout the building height. At the same time their reinforcement quantities may vary along the structural height without any restrictions. To ensure construction convenience, the design retains identical cross-sectional dimensions for all beams on the same axis.

3.1.1 JAYA Algorithm

The JAYA algorithm represents a single-phase metaheuristic optimization technique designed to progressively converge toward optimal solutions while systematically diverging from suboptimal ones. The algorithm parameters were specified with a population size of 100 individuals and a maximum iteration count of 2000 cycles. The implementation incorporates a termination criterion monitoring mechanism that records the precise iteration number at which convergence to the optimal solution occurs. Table 2 systematically presents the optimized structural solutions and their corresponding convergence iteration counts for each analytical scenario.

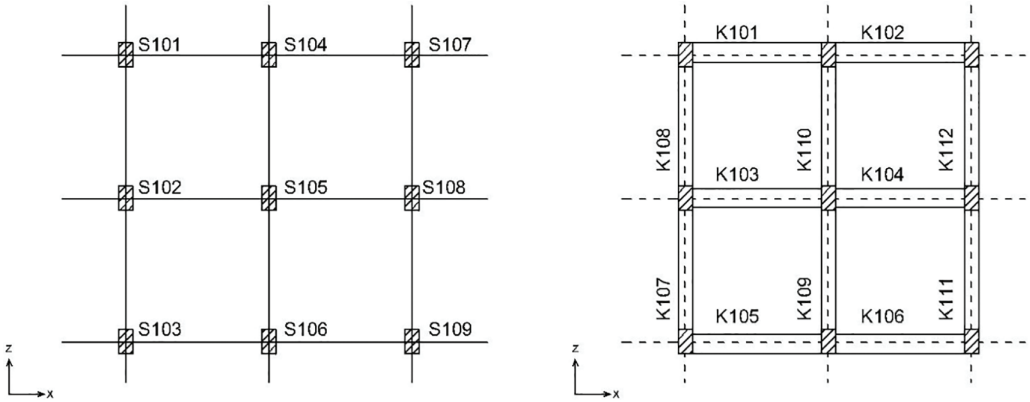

Figure 7: Column and beam layout plan of the 2-storey 2-span structure

Table 2: Cost optimization results obtained from the JAYA algorithm for the 42 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	3712.4	3712.4	3722.6	3719.9	3722.6	3712.4	3712.4	3712.4	3722.6	3722.6	3712.4	3722.6	4.90
Optimization completion step	354	256	91	94	74	135	150	107	87	86	74	87	–

The Jaya algorithm achieves global optimum solutions in 5 out of 10 conducted analyses. Fig. 8 displays the optimal cost distribution obtained from these analyses. Results indicate that the Jaya algorithm is less likely to reach global optimum solutions than other employed algorithms.

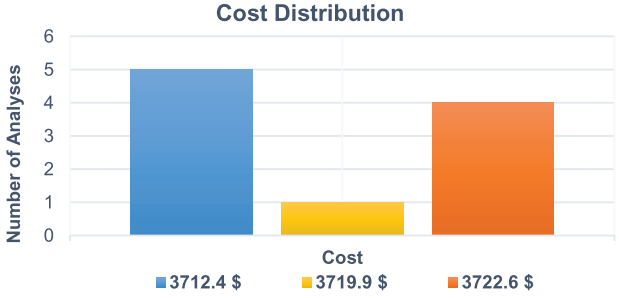

Figure 8: Distribution of analysis results from the Jaya algorithm

The convergence behavior of Analysis 8, which achieved the fastest optimization in the Jaya algorithm, is shown in Fig. 9. It is seen that the algorithm reaches the optimum result at the 107th iteration.

3.1.2 Flower Pollination Algorithm (FPA)

Table 3 presents the flower pollination algorithm (FPA) optimization performance, a two-phase metaheuristic approach. It details both the optimal solutions and their respective convergence iterations. The analysis employed a population size of 100 and a maximum of 2000 iterations, with an implemented termination criterion that recorded the exact iteration number of optimization completion.

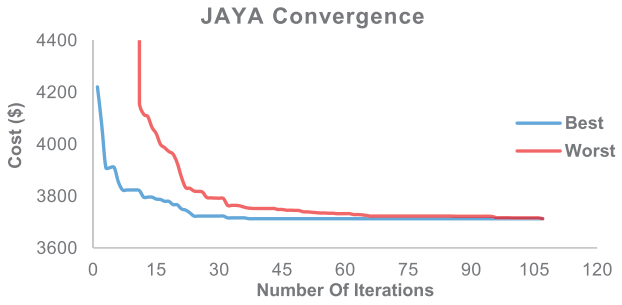

Figure 9: Convergence graph of the Jaya algorithm

Table 3: Cost optimization results obtained from the FPA algorithm for the 42-element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	3730.9	3712.4	3712.4	3712.4	3726.8	3719.8	3712.4	3748.1	3712.4	3712.4	3712.4	3748.1	12.02
Optimization completion step	105	154	331	285	118	126	180	233	218	317	126	233	–

The FPA algorithm achieves the global optimum in 6 out of 10 conducted analyses. Fig. 10 presents the optimal cost distribution obtained from these analyses.

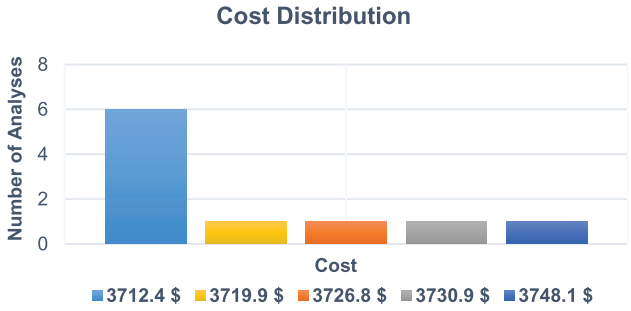

Figure 10: Distribution of analysis results from the FPA algorithm

The convergence behavior of Analysis 2 which achieved the fastest optimization in the flower pollination algorithm is shown in Fig. 11. It is seen that the algorithm reaches the optimum result at the 154th iteration.

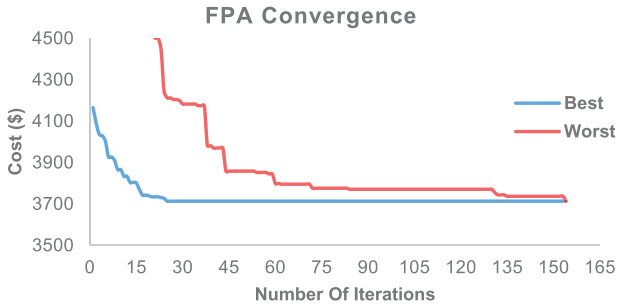

Figure 11: Convergence graph of the FPA algorithm

3.1.3 Teaching-Learning-Based Optimization (TLBO) Algorithm

The Teaching-Learning-Based Optimization (TLBO) algorithm a two-phase metaheuristic method, produces optimal solutions with their respective convergence iterations as presented in Table 4. The analysis uses a population size of 100 and runs for 2000 iterations with specialized code tracking and storing the exact iteration where optimization reaches completion.

Table 4: Cost optimization results obtained from the TLBO algorithm for the 42 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	3712.4	3713.4	3712.4	3712.4	3712.4	3719.9	3712.4	3712.4	3712.4	3712.4	3712.4	3719.9	2.36
Optimization completion step	56	41	73	71	66	88	61	58	103	70	58	88	–

The TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 12 presents the optimal cost distribution obtained from these analyses.

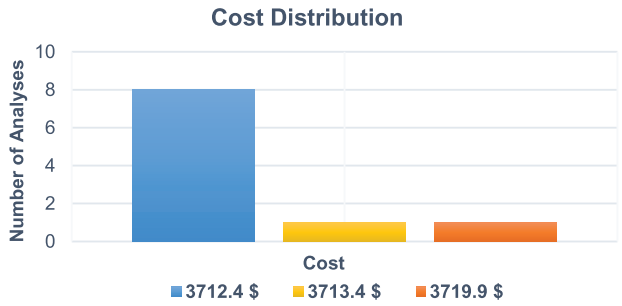

Figure 12: Distribution of analysis results from the TLBO algorithm

The convergence behavior of Analysis 1 which achieved the fastest optimization in the Teaching-Learning-Based Optimization (TLBO) algorithm is shown in Fig. 13. It is seen that the algorithm reaches the optimum result at the 56th iteration.

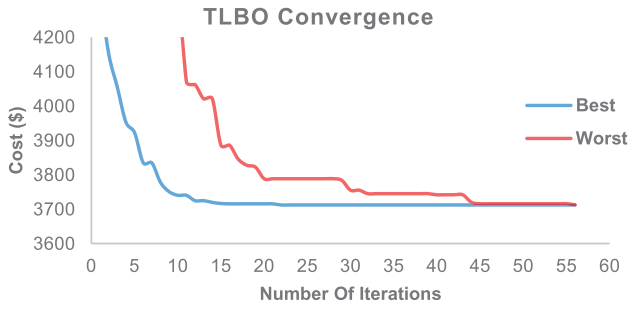

Figure 13: Convergence graph of the TLBO algorithm

3.1.4 JA/FPA Hybrid Algorithm

The flower pollination algorithm is a two-phase algorithm. In the first phase, it performs a global search, while in the second phase, it conducts a local search. The algorithm uses the Jaya algorithm in its first phase and retains the original second phase of the flower pollination algorithm. The number

of iterations is set to 2000 with implemented code tracking the exact iteration step where optimization concludes and saving this completion step. The population size is taken as 100. The ps factor, which serves as a switching probability between global and local search, is used in the calculations as shown below. The ps factor has been set to 0.5 to provide equal opportunity for both types of optimization.

If ps < 0.5 Phase 1

If ps > 0.5 Phase 2 continues.

The static analysis of the frame system involves numerous variables and varying reinforcement quantities for each section, which may cause the algorithm to converge to local optima. Therefore, each analysis was repeated 10 times. Table 5 presents the optimal solution obtained for the structure and the iteration number at which the algorithm reached the optimum result in each analysis.

Table 5: Cost optimization results obtained from the JA/FPA algorithm for the 42 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	3712.4	3712.4	3712.4	3712.4	3791.2	3712.4	3712.4	3712.4	3712.4	3724.0	3712.4	3791.2	24.78
Optimization completion step	410	807	824	102	117	141	144	746	113	78	746	117	–

The hybrid algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 14 presents the optimal cost distribution obtained from these analyses.

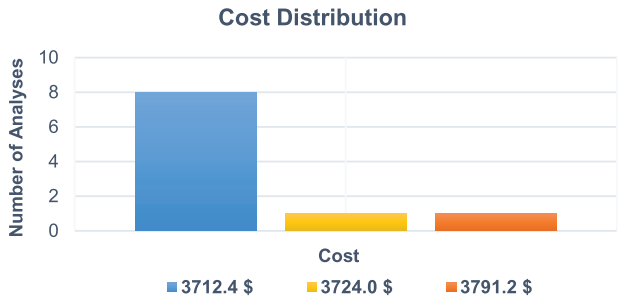

Figure 14: Distribution of analysis results from the JA/FPA algorithm

The convergence behavior of Analysis 4 which achieved the fastest optimization in the JA/FPA algorithm is shown in Fig. 15. It is observed that the algorithm achieves the optimum result at the 102nd iteration.

3.1.5 JA/TLBO Hybrid Algorithm

The teaching-learning-based optimization algorithm is a two-phase algorithm. The new hybrid algorithm is similar to the previous hybrid algorithm, with the only difference being that no ps factor is used in this algorithm. In each iteration of the hybrid algorithm, the Jaya algorithm is used first, followed by the student phase of the TLBO algorithm sequentially.

The population size is set to 100. The number of iterations is specified as 2000 with implemented code tracking the exact iteration where optimization concludes and recording this completion step. Each analysis is repeated 10 times. Table 6 presents the optimal solution achieved for the structure and the iteration number at which the algorithm reaches the optimum result in each analysis.

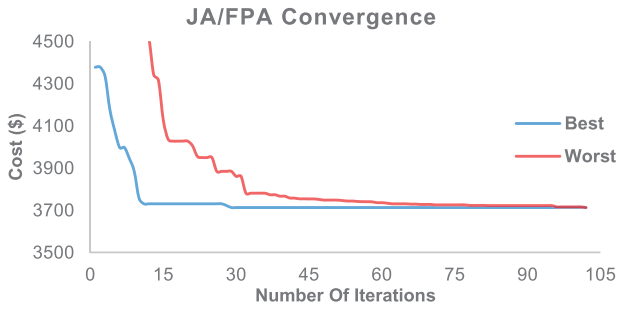

Figure 15: Convergence graph of the JA/FPA algorithm

Table 6: Cost optimization results obtained from the JA/TLBO algorithm for the 42 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	3712.4	3722.6	3712.4	3712.4	3712.4	3712.4	3712.4	3712.4	3722.6	3712.4	3712.4	3722.6	4.30
Optimization completion step	80	66	168	319	91	330	77	114	73	102	114	73	–

The JA/TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 16 presents the optimal cost distribution obtained from these analyses.

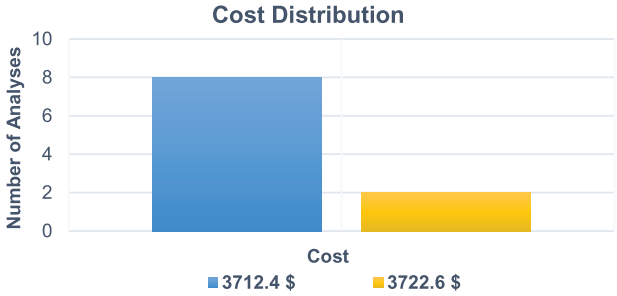

Figure 16: Distribution of analysis results from the JA/TLBO algorithm

The convergence behavior of Analysis 7 which achieved the fastest optimization in the JA/TLBO algorithm is shown in Fig. 17. It is seen that the algorithm reaches the optimum result at the 77th iteration.

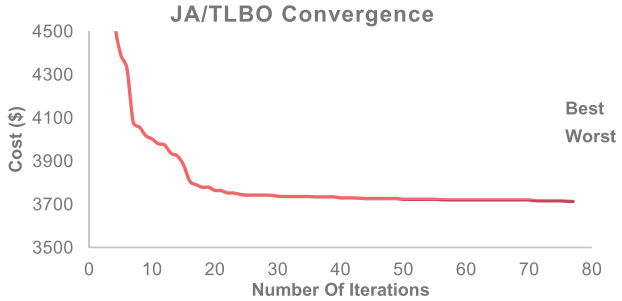

Figure 17: Convergence graph of the JA/TLBO algorithm

This study employs five different optimization algorithms, including two hybrid algorithms. These five optimization algorithms perform static calculations and analyses of the RC frame system. Each algorithm solves the structure with 10 repetitions, and the results are recorded. Table 7 presents the optimal results and standard deviation values of the algorithms. One-way ANOVA test was conducted for these results and p-value is found as 0.56. Since $p > 0.05$, there is no statistically significant difference.

Table 7: Optimum cost results and standard deviation values calculated by the algorithms

Algorithm	1	2	3	4	5	6	7	8	9	10	Standard deviation
JAYA COST ($)	3712.4	3712.4	3722.6	3719.9	3722.6	3712.4	3712.4	3712.4	3722.6	3722.6	4.90
FPA COST ($)	3730.9	3712.4	3712.4	3712.4	3726.8	3719.8	3712.4	3748.1	3712.4	3712.4	12.02
TLBO COST ($)	3712.4	3713.4	3712.4	3712.4	3712.4	3719.9	3712.4	3712.4	3712.4	3712.4	2.36
JA/FPA COST ($)	3712.4	3712.4	3712.4	3712.4	3791.2	3712.4	3712.4	3712.4	3712.4	3724.0	24.78
JA/TLBO COST ($)	3712.4	3722.6	3712.4	3712.4	3712.4	3712.4	3712.4	3712.4	3722.6	3712.4	4.30

All algorithms reach optimal solutions but when examining stability, faster convergence, and standard deviation values the Teaching-Learning-Based Optimization (TLBO) algorithm demonstrates superior performance in this case. The TLBO algorithm achieves the best standard deviation value at 2.36 while the JA/FPA hybrid algorithm shows the worst performance with a standard deviation of 24.78. Table 8 presents the section dimensions and each element's cost for the 42-member RC space frame system based on the analysis results.

Table 8: Section dimensions and optimum cost values of the 42-element RC system obtained using the TLBO

Element name	b_w (m)	h (m)	Cost ($)
S101	0.30	0.30	54.60
S102	0.45	0.30	72.02
S103	0.30	0.30	54.60
S104	0.35	0.30	76.55
S105	0.40	0.40	98.70
S106	0.35	0.30	76.55
S107	0.30	0.30	54.60
S108	0.45	0.30	72.02
S109	0.30	0.30	54.60
K101	0.30	0.55	99.08
K102	0.30	0.55	99.08

(Continued)

Table 8 (continued)

Element name	b_w (m)	h (m)	Cost ($)
K103	0.30	0.55	98.53
K104	0.30	0.55	98.53
K105	0.30	0.55	99.08
K106	0.30	0.55	99.08
K107	0.30	0.60	101.63
K108	0.30	0.60	101.63
K109	0.30	0.60	100.70
K110	0.30	0.60	100.70
K111	0.30	0.60	101.63
K112	0.30	0.60	101.63
S201	0.30	0.30	76.12
S202	0.45	0.30	61.26
S203	0.30	0.30	76.12
S204	0.35	0.30	84.91
S205	0.40	0.40	73.20
S206	0.35	0.30	84.91
S207	0.30	0.30	76.12
S208	0.45	0.30	61.26
S209	0.30	0.30	76.12
K201	0.30	0.60	103.35
K202	0.30	0.60	103.35
K203	0.30	0.60	102.81
K204	0.30	0.60	102.81
K205	0.30	0.60	103.35
K206	0.30	0.60	103.35
K207	0.30	0.55	101.45
K208	0.30	0.55	101.45
K209	0.30	0.55	100.98
K210	0.30	0.55	100.98
K211	0.30	0.55	101.45
K212	0.30	0.55	101.45
Total Cost			**3712.4$**

The RC frame system consisting of 42 elements undergoes static analysis in SAP2000 using the optimal section dimensions obtained through a metaheuristic algorithm-based design. The results from MATLAB and SAP2000 are compared, demonstrating the analysis' validity. Table 9 displays displacement values from both MATLAB and SAP2000 for four selected structural elements (one column and one beam per floor).

Table 9: Comparison of displacement values obtained from MATLAB and SAP2000

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S104(i)	0	0	0	0	0	0
MATLAB		0	0	0	0	0	0
SAP2000	S104(j)	0	−0.000029	−0.00305	−0.00127	0	0
MATLAB		0	−0.000029	−0.00305	0.00123	0	0
SAP2000	K101(i)	−0.000023	−0.000023	−0.00192	−0.00161	0.00207	−0.000000078
MATLAB		−0.000023	−0.000022	−0.00192	0.00155	−0.00200	0.000000014
SAP2000	K101(j)	0	−0.000029	−0.00305	−0.00127	0	0
MATLAB		0	−0.000029	−0.00305	0.00123	0	0
SAP2000	S205(i)	0	0	−0.00295	0	0	0
MATLAB		0	0	−0.00297	0	0	0
SAP2000	S205(j)	0	0	−0.00446	0	0	0
MATLAB		0	0	−0.00447	0	0	0
SAP2000	K206(i)	0	−0.000053	−0.00458	0.00198	0	0
MATLAB		0	−0.000054	−0.00458	0.00195	0	0
SAP2000	K206(j)	−0.000042	−0.0000420	−0.00287	0.00239	−0.00226	0.00000045
MATLAB		0.000042	−0.0000417	−0.00287	0.00230	0.00218	0.00000054

The evaluation of displacement results obtained from MATLAB and SAP2000 analyses using the Mean Absolute Error (MAE) method is presented in Table 10. A comparison of values obtained from the code developed based on the Mean Absolute Error (MAE) method and the SAP2000 software revealed that the maximum difference in displacement values was 0.002 cm, while the maximum difference in rotation values was 0.00009 radians. The differences between the results presented in Table 10 have been determined to originate from the rounding errors used by the programs in their calculations. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 10: MAE comparison of displacement values: MATLAB vs. SAP2000

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000	S104(i)	0	0	0	0	0	0
MATLAB							
SAP2000	S104(j)	0	0	0	0.00004	0	0
MATLAB							
SAP2000	K101(i)	0	0.000001	0	0.00006	0.00007	0.000000064
MATLAB							
SAP2000	K101(j)	0	0	0	0.00004	0	0
MATLAB							

(Continued)

Table 10 (continued)

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000 MATLAB	S205(i)	0	0	0.00002	0	0	0
SAP2000 MATLAB	S205(j)	0	0	0.00001	0	0	0
SAP2000 MATLAB	K206(i)	0	0.000001	0	0.00003	0	0
SAP2000 MATLAB	K206(j)	0	0.0000003	0	0.00009	0.00008	0.00000009

3.2 Example 2

This section addresses optimizing a two-story RC frame system with two spans in the z-direction and three spans in the x-direction. The structure contains 24 columns and 34 beams, totaling 58 structural elements. The design features 4-m span lengths and 3-m story heights. Fig. 18 illustrates a view of the frame system, loading conditions, and cross-section along the x-axis. The structure carries a 100 kN/m dead load and 50 kN/m live load with load combinations using only vertical loads (1.4G + 1.6Q) applied.

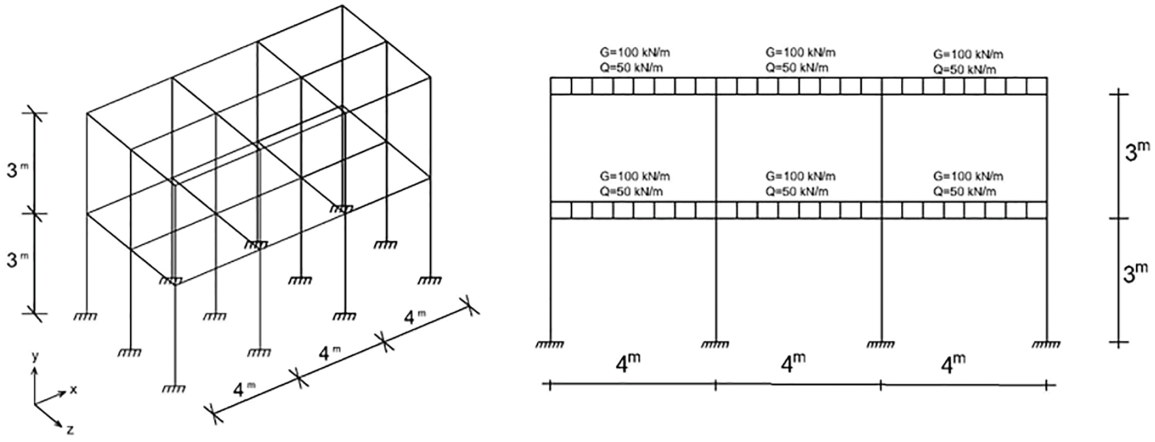

Figure 18: Load distribution and cross-section view of the 2-storey 3-span structure

Fig. 19 presents the floor plan showing columns and beams for the RC frame system illustrated in Fig. 18. The design maintains constant cross-sectional areas for all columns throughout the building height while allowing variation in reinforcement quantities along the columns. For beams, the design ensures uniform cross-sectional dimensions for all beams along the same axis to facilitate construction practicality.

3.2.1 Jaya Algorithm

The Jaya algorithm is a single-phase optimization method that aims to approach the best solution while avoiding poor solutions. The analysis uses a population size of 100 and 2000 iterations, with implemented code tracking the exact iteration where optimization concludes and recording this

completion step. Table 11 presents both the optimal structural solution and the iteration number at which the algorithm reaches the optimum result for each analysis.

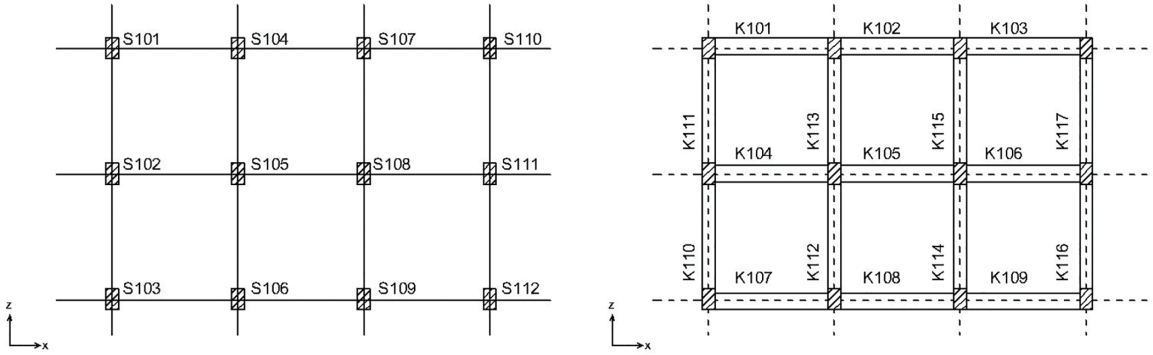

Figure 19: Column and beam layout plan of the 2-storey 3-span structure

Table 11: Cost optimization results obtained from the JAYA algorithm for the 58 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	5116.8	5118.6	5131.1	5118.6	5118.6	5116.8	5135.9	5118.6	5118.6	5150.1	5116.8	5150.1	11.60
Optimization completion step	128	89	76	108	107	143	149	82	86	78	89	78	–

Fig. 20 displays the optimal cost distribution obtained from the analyses. The Jaya algorithm demonstrates a lower probability of reaching global optimum solutions compared to other algorithms used. Results show the Jaya algorithm achieves global optimum in only 2 out of 10 conducted analyses.

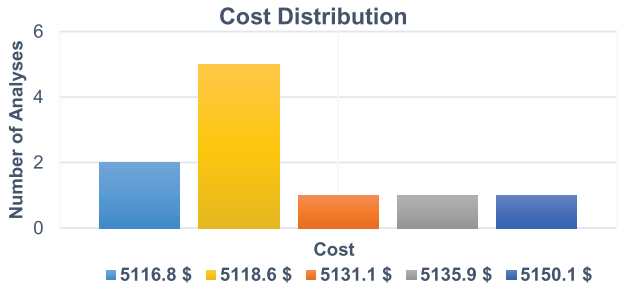

Figure 20: Distribution of analysis results from the Jaya algorithm

The convergence behavior of Analysis 1 which achieved the fastest optimization in the Jaya algorithm is shown in Fig. 21. It is seen that the algorithm reaches the optimum result at the 128th iteration.

3.2.2 Flower Pollination Algorithm (FPA)

Table 12 below shows the optimal solutions and the iteration numbers reaching optimum results for the flower pollination algorithm, a two-phase metaheuristic algorithm. The population size was set to 100 and iterations to 2000, with code implemented to track and record the exact iteration step where optimization completed.

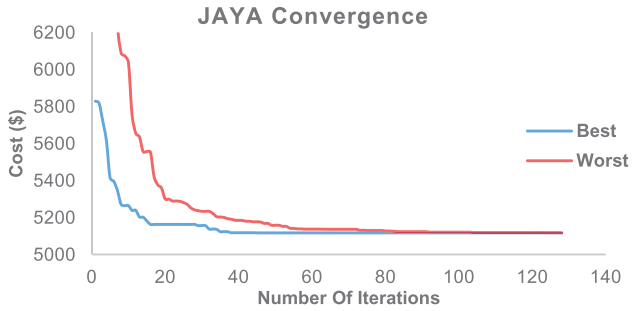

Figure 21: Convergence graph of the Jaya algorithm

Table 12: Cost optimization results obtained from the FPA algorithm for the 58 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	5118.6	5136.3	5116.8	5136.3	5116.8	5116.8	5130.9	5116.8	5116.8	5170.1	5116.8	5170.1	16.85
Optimization completion step	118	117	400	149	146	147	133	128	174	150	146	150	–

The FPA algorithm achieves the global optimum in 5 out of 10 conducted analyses. Fig. 22 presents the optimal cost distribution obtained from these analyses.

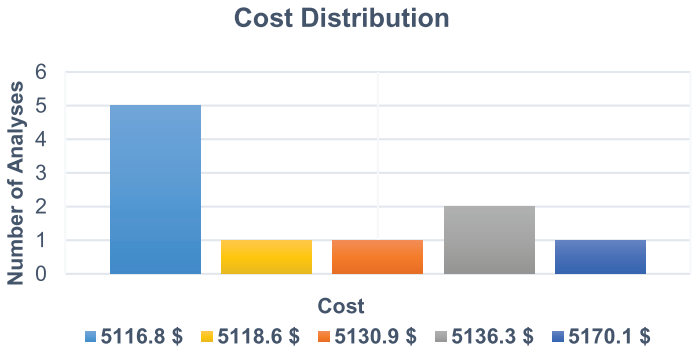

Figure 22: Distribution of analysis results from the FPA algorithm

The convergence behavior of Analysis 8 which achieved the fastest optimization in the FPA is shown in Fig. 23. It is seen that the algorithm reaches the optimum result at the 128th iteration.

3.2.3 Teaching-Learning-Based Optimization (TLBO) Algorithm

The Teaching-Learning-Based Optimization (TLBO) algorithm, a two-phase metaheuristic method, produces optimal solutions with their respective convergence iterations as presented in Table 13. The analysis uses a population size of 100 and runs for 2000 iterations with specialized code tracking and storing the exact iteration where optimization reaches completion.

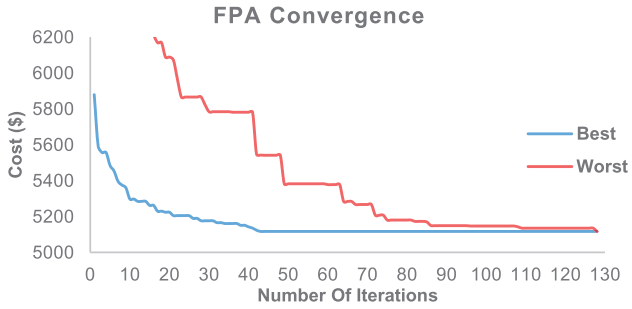

Figure 23: Convergence graph of the FPA algorithm

Table 13: Cost optimization results obtained from the TLBO algorithm for the 58 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5118.6	5116.8	5118.6	0.57
Optimization completion step	37	46	57	43	31	77	49	67	87	51	87	51	–

The TLBO algorithm achieves the global optimum in 9 out of 10 conducted analyses. Fig. 24 presents the optimal cost distribution obtained from these analyses.

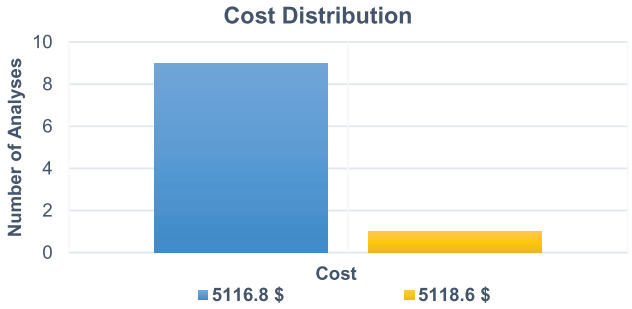

Figure 24: Distribution of analysis results from the TLBO algorithm

The convergence behavior of Analysis 5 which achieved the fastest optimization in the TLBO algorithm is shown in Fig. 25. It is seen that the algorithm reaches the optimum result at the 31th iteration.

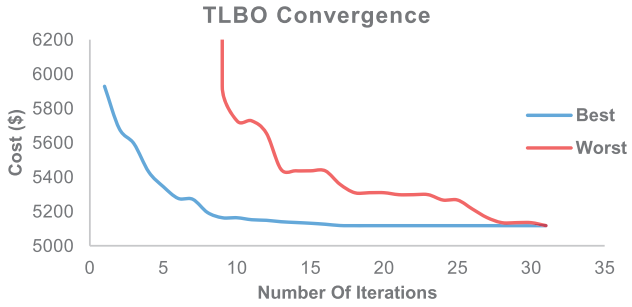

Figure 25: Convergence graph of the TLBO algorithm

3.2.4 JA/FPA Hybrid Algorithm

The flower pollination algorithm is a two-phase algorithm. In the first phase, it performs a global search in the second phase, it conducts a local search. The algorithm uses the Jaya algorithm in its first phase and retains the original second phase of the flower pollination algorithm. The number of iterations is set to 2000 with implemented code tracking the exact iteration step where optimization concludes and saving this completion step. The population size is taken as 100. The ps factor which serves as a switching probability between global and local search is used in the calculations as shown below. The ps factor has been set to 0.5 to provide equal opportunity for both types of optimization.

If ps < 0.5 Phase 1

If ps > 0.5 Phase 2 continues.

The static analysis of the frame system involves numerous variables and varying reinforcement quantities for each section, which may cause the algorithm to converge to local optima. Therefore, each analysis was repeated 10 times. Table 14 presents the optimal solution obtained for the structure and the iteration number at which the algorithm reached the optimum result in each analysis.

Table 14: Cost optimization results obtained from the JA/FPA algorithm for the 58 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5136.3	5116.8	5116.8	5116.8	5136.3	6.16
Optimization completion step	149	235	196	168	173	179	148	108	214	186	214	108	–

The hybrid algorithm achieves the global optimum in 9 out of 10 conducted analyses. Fig. 26 presents the optimal cost distribution obtained from these analyses.

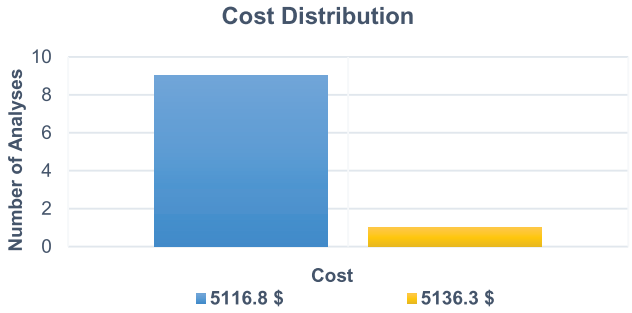

Figure 26: Distribution of analysis results from the JA/FPA algorithm

The convergence behavior of Analysis 7, which achieved the fastest optimization in the JA/FPA algorithm, is shown in Fig. 27. It is seen that the algorithm reaches the optimum result at the 148th iteration.

3.2.5 JA/TLBO Hybrid Algorithm

The teaching-learning-based optimization algorithm is a two-phase algorithm. The new hybrid algorithm is similar to the previous one, with the only difference being that no ps factor is used in this algorithm. In each iteration of the hybrid algorithm, the Jaya algorithm is used first, followed by the student phase of the TLBO algorithm sequentially.

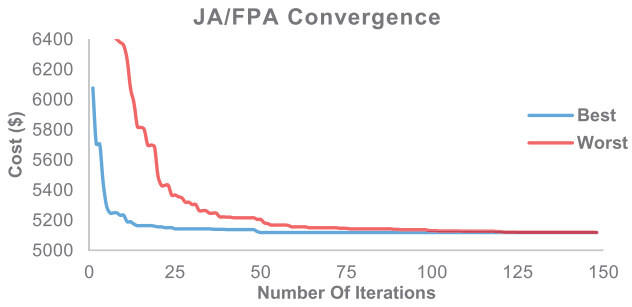

Figure 27: Convergence graph of the JA/FPA algorithm

The population size is set to 100. The number of iterations is specified as 2000 with implemented code tracking the exact iteration where optimization concludes and recording this completion step. Each analysis is repeated 10 times. Table 15 presents the optimal solution achieved for the structure and the iteration number at which the algorithm reaches the optimum result in each analysis.

Table 15: Cost optimization results obtained from the JA/TLBO algorithm for the 58 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5118.6	5116.8	5136.3	5116.8	5116.8	5136.3	6.14
Optimization completion step	82	102	97	78	103	91	92	88	78	105	88	78	–

The JA/TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 28 presents the optimal cost distribution obtained from these analyses.

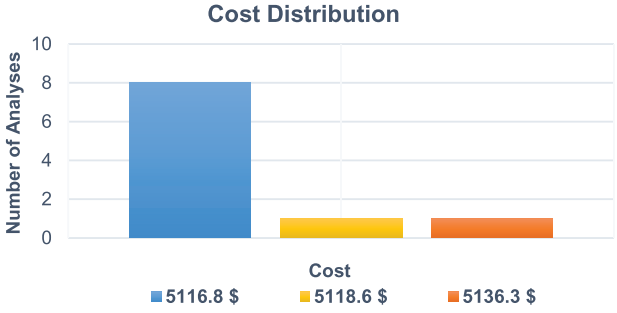

Figure 28: Distribution of analysis results from the JA/TLBO algorithm

The convergence behavior of Analysis 4 which achieved the fastest optimization in the JA/TLBO algorithm is shown in Fig. 29. It is seen that the algorithm reaches the optimum result at the 78th iteration.

This study utilizes five different optimization algorithms, including two hybrid algorithms. These five algorithms perform static calculations and optimization analyses for the RC frame system. Each algorithm solves the structure with 10 repetitions, and the results are recorded. Table 16 presents the optimal results and standard deviation values of the algorithms. One-way ANOVA test has a p-value of 0.106 and there is no statistically significant difference between the algorithms.

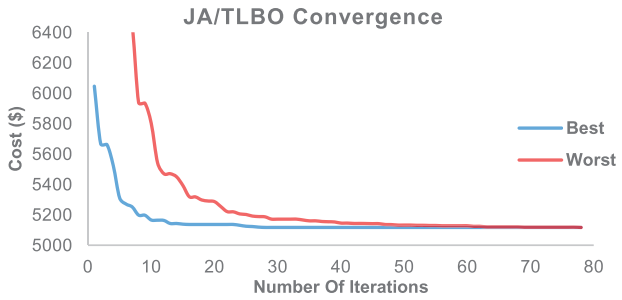

Figure 29: Convergence graph of the JA/TLBO algorithm

Table 16: Optimum cost results and standard deviation values calculated by the algorithms

Algorithm	1	2	3	4	5	6	7	8	9	10	Standard deviation
JAYA COST ($)	5116.8	5118.6	5131.1	5118.6	5118.6	5116.8	5135.9	5118.6	5118.6	5150.1	11.16
FPA COST ($)	5118.6	5136.3	5116.8	5136.3	5116.8	5116.8	5130.9	5116.8	5116.8	5170.1	16.85
TLBO COST ($)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5118.6	0.57
JA/FPA COST ($)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5136.3	5116.8	5116.8	6.16
JA/TLBO COST ($)	5116.8	5116.8	5116.8	5116.8	5116.8	5116.8	5118.6	5116.8	5136.3	5116.8	6.14

All algorithms achieve optimal solutions, but when evaluating stability, convergence speed, and standard deviation values, the Teaching-Learning-Based Optimization (TLBO) algorithm demonstrates superior performance in this case. The analysis reveals the Flower Pollination Algorithm (FPA) shows the poorest convergence with a standard deviation value of 16.85. Table 17 displays the section dimensions and individual element costs for the 58-member RC space frame system based on the analysis results.

Table 17: Section dimensions and optimum cost values of the 58-element RC system obtained using the TLBO

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S101	0.30	0.30	54.60	S201	0.30	0.30	76.12
S102	0.45	0.30	72.02	S202	0.45	0.30	66.64
S103	0.30	0.30	54.60	S203	0.30	0.30	76.12
S104	0.35	0.30	68.18	S204	0.35	0.30	84.91
S105	0.45	0.35	90.80	S205	0.45	0.35	71.97
S106	0.35	0.30	68.18	S206	0.35	0.30	84.91
S107	0.35	0.30	68.18	S207	0.35	0.30	84.91

(Continued)

Table 17 (continued)

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S108	0.45	0.35	90.80	S208	0.45	0.35	71.97
S109	0.35	0.30	68.18	S209	0.35	0.30	84.91
S110	0.30	0.30	54.60	S210	0.30	0.30	76.12
S111	0.45	0.30	72.02	S211	0.45	0.30	66.64
S112	0.30	0.30	54.60	S212	0.30	0.30	76.12
K101	0.30	0.50	98.14	K201	0.30	0.50	99.78
K102	0.30	0.50	93.60	K202	0.30	0.50	93.22
K103	0.30	0.50	98.14	K203	0.30	0.50	99.78
K104	0.30	0.55	98.82	K204	0.30	0.60	103.89
K105	0.30	0.55	94.57	K205	0.30	0.60	96.25
K106	0.30	0.55	98.82	K206	0.30	0.60	103.89
K107	0.30	0.50	98.14	K207	0.30	0.50	99.78
K108	0.30	0.50	93.60	K208	0.30	0.50	93.22
K109	0.30	0.50	98.14	K209	0.30	0.50	99.78
K110	0.30	0.60	101.67	K210	0.30	0.55	101.56
K111	0.30	0.60	101.67	K211	0.30	0.55	101.56
K112	0.30	0.60	100.49	K212	0.30	0.55	100.58
K113	0.30	0.60	100.49	K213	0.30	0.55	100.58
K114	0.30	0.60	100.49	K214	0.30	0.55	100.58
K115	0.30	0.60	100.49	K215	0.30	0.55	100.58
K116	0.30	0.60	101.67	K216	0.30	0.55	101.56
K117	0.30	0.60	101.67	K217	0.30	0.55	101.56
		Total Cost				**5116.8$**	

The RC frame system with 58 elements is analyzed in SAP2000 using optimal section dimensions obtained through metaheuristic algorithm optimization. The MATLAB and SAP2000 results are compared to confirm the accuracy of the analysis. Table 18 shows displacement values from both MATLAB and SAP2000 for four structural elements selected as one column and one beam from each floor.

Table 18: Comparison of displacement values obtained from MATLAB and SAP2000

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S101(i)	0	0	0	0	0	0
MATLAB		0	0	0	0	0	0
SAP2000	S101(j)	−0.000050	−0.000023	−0.00196	−0.001619	0.00237	−0.00000201
MATLAB		−0.000049	−0.000025	−0.00196	0.001555	−0.00229	0.00000178
SAP2000	K109(i)	0.000014	0.000029	−0.00289	0.001287	0.00031	−0.00000087
MATLAB		0.000014	0.000029	−0.00288	−0.001237	−0.00031	0.00000088

(Continued)

Table 18 (continued)

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	K109(j)	0.000050	0.000023	−0.00196	0.001619	−0.00237	−0.00000201
MATLAB		0.000049	0.000025	−0.00196	−0.001555	0.00229	0.00000178
SAP2000	S205(i)	−0.000011	0	−0.00289	0	−0.00023	0
MATLAB		−0.000011	0	−0.00290	0	0.00024	0
SAP2000	S205(j)	0.00002	0	−0.00435	0	−0.00035	0
MATLAB		0.00002	0	−0.00436	0	0.00038	0
SAP2000	K204(i)	0.000075	0	−0.00364	0	0.00205	0
MATLAB		−0.000076	0	−0.00364	0	−0.00205	0
SAP2000	K204(j)	0.00002	0	−0.00435	0	−0.00035	0
MATLAB		0.00002	0	−0.00436	0	0.00038	0

The evaluation of displacement results obtained from MATLAB and SAP2000 analyses using the Mean Absolute Error (MAE) method is presented in Table 19. A comparison of values obtained from the code developed based on the Mean Absolute Error (MAE) method and the SAP2000 software revealed that the maximum difference in displacement values was 0.001 cm, while the maximum difference in rotation values was 0.00008 radians. The differences between the results presented in Table 19 have been determined to originate from the rounding errors used by the programs in their calculations. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 19: MAE comparison of displacement values: MATLAB vs. SAP2000

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000	S101(i)	0	0	0	0	0	0
MATLAB							
SAP2000	S101(j)	0.000001	0.000002	0	0.000064	0.00008	0.00000023
MATLAB							
SAP2000	K109(i)	0	0	0.00001	0.00005	0	0.00000001
MATLAB							
SAP2000	K109(j)	0.000001	0.000002	0	0.000064	0.00008	0.00000023
MATLAB							
SAP2000	S205(i)	0	0	0.00001	0	0.00001	0
MATLAB							
SAP2000	S205(j)	0	0	0.00001	0	0.00003	0
MATLAB							
SAP2000	K204(i)	0.000001	0	0	0	0	0
MATLAB							
SAP2000	K204(j)	0	0	0.00001	0	0.00003	0
MATLAB							

When examining the optimal results from all algorithms, the Teaching-Learning-Based Optimization (TLBO) algorithm demonstrates superior performance in optimizing space RC frame systems, considering stability, faster convergence, and standard deviation values. In subsequent structural optimization solutions, only the TLBO algorithm was employed for the analysis.

3.3 Example 3

This section examines optimizing a five-story RC frame system with two spans along the z-axis and three spans along the x-axis. The structure contains 60 columns and 85 beams, totaling 145 structural members. The design features 4-m span lengths and 3-m story heights. Fig. 30 illustrates a view of the frame system, loading conditions, and cross-section along the x-axis. The structure carries a 100 kN/m dead load and 50 kN/m live load with load combinations using only vertical loads (1.4G + 1.6Q) applied.

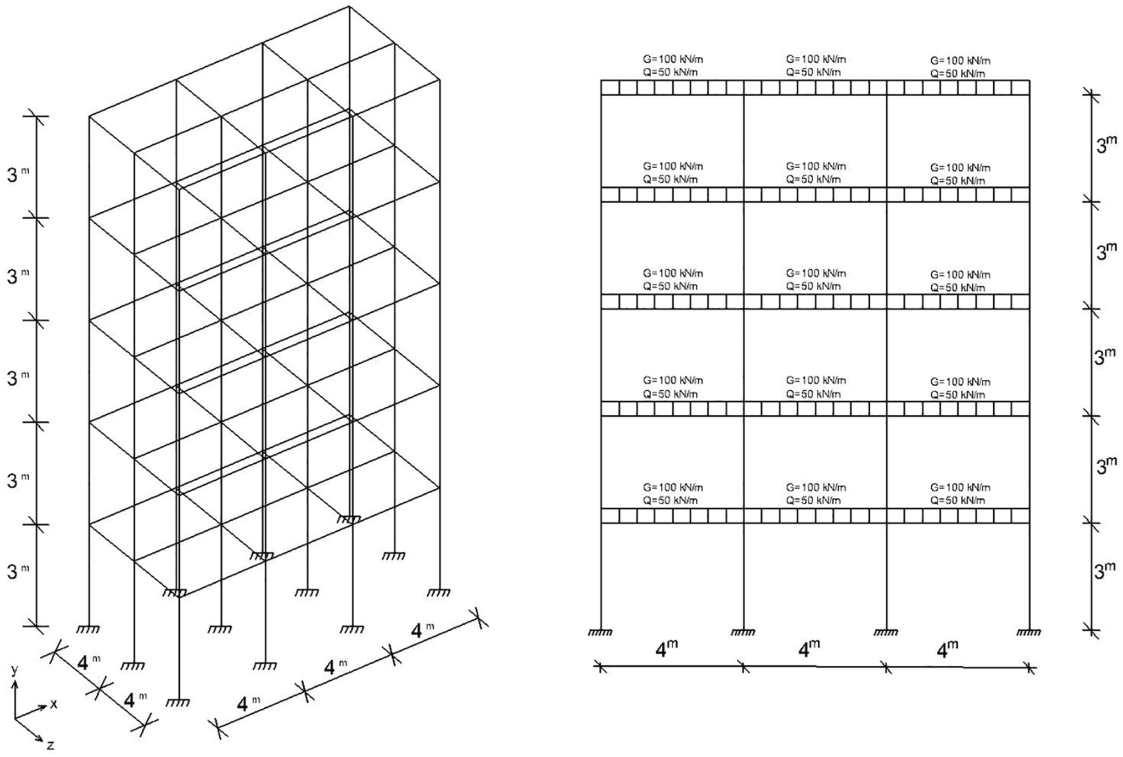

Figure 30: Load distribution and cross-section view of the 5-storey 3-span structure

Fig. 31 displays the floor plan showing column and beam layout for the RC frame system presented in Fig. 30. The design maintains constant cross-sectional areas for columns throughout the building height while allowing variable reinforcement quantities along columns. For beams, the design enforces uniform cross-sectional dimensions for all beams along the same axis to ensure construction efficiency.

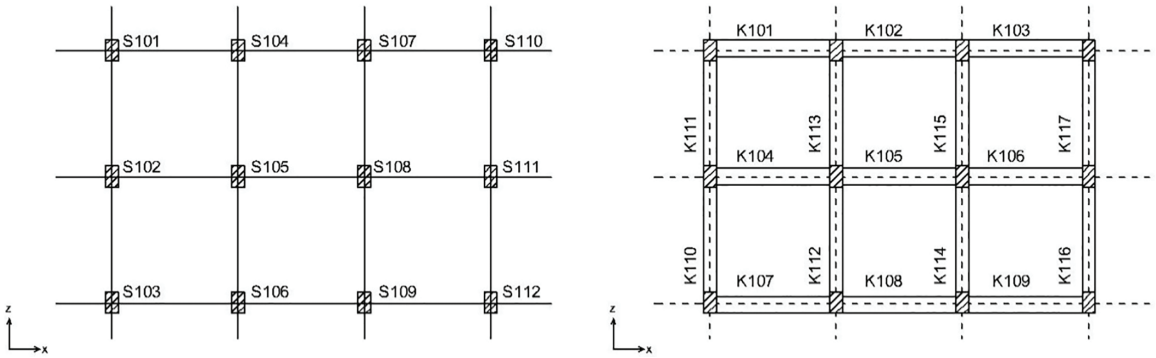

Figure 31: Column and beam layout plan of the 5-storey 3-span structure

The Teaching-Learning-Based Optimization (TLBO) algorithm, a two-phase metaheuristic method, produces optimal solutions with their respective convergence iterations as presented in Table 20.

Table 20: Cost optimization results obtained from the TLBO algorithm for the 145 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	17,042.4	17,043.5	17,042.4	17,043.5	17,042.4	17,042.4	17,042.4	17,042.4	17,042.4	17,042.4	17,042.4	17,043.5	0.50
Optimization completion step	70	81	77	198	63	92	93	71	62	65	71	198	–

Previous sections analyzed two different frame systems with 42 and 58 elements. A population size of 100 and 2000 iterations is utilized, with implemented code tracking and recording the exact iteration step where optimization completes.

The TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 32 presents the optimal cost distribution obtained from these analyses.

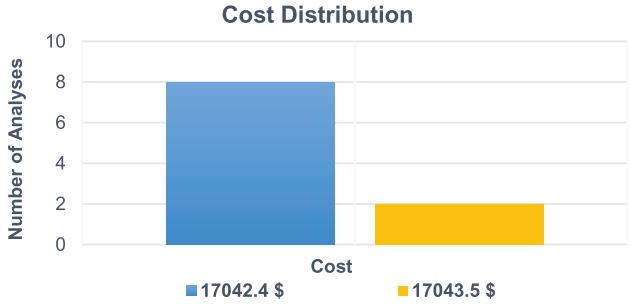

Figure 32: Distribution of analysis results from the TLBO algorithm

The convergence behaviour of Analysis 9 which achieved the fastest optimization in the TLBO algorithm is shown in Fig. 33. It is seen that the algorithm reaches the optimum result at the 62th iteration

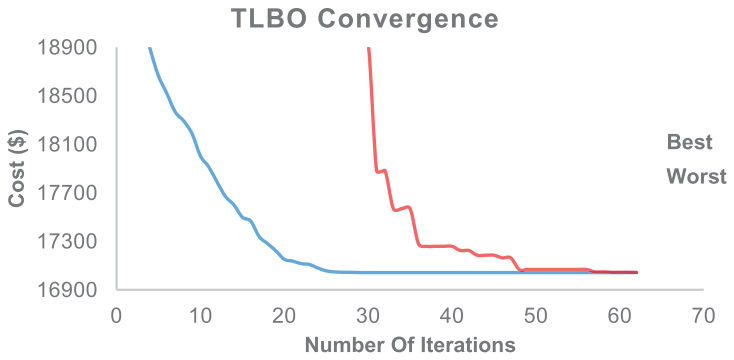

Figure 33: Convergence graph of the TLBO algorithm

According to the analysis results, Table 21 displays the cross-sectional dimensions and individual element costs for the 145-member RC space frame system.

Table 21: Section dimensions and optimum cost values of the 145-element RC system

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S101	0.50	0.30	116.08	K209	0.30	0.55	107.08	S412	0.50	0.30	77.83
S102	0.50	0.55	163.02	K210	0.30	0.60	107.39	K401	0.30	0.55	107.24
S103	0.50	0.30	116.08	K211	0.30	0.60	107.59	K402	0.30	0.55	107.05
S104	0.50	0.45	204.50	K212	0.30	0.60	107.59	K403	0.30	0.55	107.24
S105	0.50	0.60	264.23	K213	0.30	0.60	107.32	K404	0.30	0.55	107.03
S106	0.50	0.45	204.50	K214	0.30	0.60	107.32	K405	0.30	0.55	107.02
S107	0.50	0.45	204.50	K215	0.30	0.60	107.32	K406	0.30	0.55	107.03
S108	0.50	0.60	264.23	K216	0.30	0.60	107.32	K407	0.30	0.55	107.24
S109	0.50	0.45	204.50	K217	0.30	0.60	107.59	K408	0.30	0.55	107.05
S110	0.50	0.30	116.08	S301	0.50	0.30	107.59	K409	0.30	0.55	107.24
S111	0.50	0.55	163.02	S302	0.50	0.55	82.61	K410	0.30	0.60	107.53
S112	0.50	0.30	116.08	S303	0.50	0.30	136.72	K411	0.30	0.60	107.53
K101	0.30	0.55	107.63	S304	0.50	0.45	82.61	K412	0.30	0.60	107.26
K102	0.30	0.55	107.11	S305	0.50	0.60	118.43	K413	0.30	0.60	107.26
K103	0.30	0.55	107.63	S306	0.50	0.45	149.46	K414	0.30	0.60	107.26
K104	0.30	0.55	106.96	S307	0.50	0.45	118.43	K415	0.30	0.60	107.26
K105	0.30	0.55	107.05	S308	0.50	0.60	118.43	K416	0.30	0.60	107.53
K106	0.30	0.55	106.96	S309	0.50	0.45	149.46	K417	0.30	0.60	107.53
K107	0.30	0.55	107.63	S310	0.50	0.30	118.43	S501	0.50	0.30	111.30
K108	0.30	0.55	107.11	S311	0.50	0.55	82.61	S502	0.50	0.55	127.96
K109	0.30	0.55	107.63	S312	0.50	0.30	136.72	S503	0.50	0.30	111.30
K110	0.30	0.60	107.66	K301	0.30	0.55	82.61	S504	0.50	0.45	154.29
K111	0.30	0.60	107.66	K302	0.30	0.55	107.32	S505	0.50	0.60	139.89
K112	0.30	0.60	107.40	K303	0.30	0.55	107.08	S506	0.50	0.45	154.29
K113	0.30	0.60	107.40	K304	0.30	0.55	107.32	S507	0.50	0.45	154.29
K114	0.30	0.60	107.40	K305	0.30	0.55	106.96	S508	0.50	0.60	139.89
K115	0.30	0.60	107.40	K306	0.30	0.55	107.04	S509	0.50	0.45	154.29
K116	0.30	0.60	107.66	K307	0.30	0.55	106.96	S510	0.50	0.30	111.30
K117	0.30	0.60	107.66	K308	0.30	0.55	107.32	S511	0.50	0.55	127.96
S201	0.50	0.30	87.39	K309	0.30	0.55	107.08	S512	0.50	0.30	111.30
S202	0.50	0.55	145.49	K310	0.30	0.60	107.32	K501	0.30	0.55	108.61
S203	0.50	0.30	87.39	K311	0.30	0.60	107.67	K502	0.30	0.55	107.17
S204	0.50	0.45	125.60	K312	0.30	0.60	107.67	K503	0.30	0.55	108.61
S205	0.50	0.60	168.59	K313	0.30	0.60	107.33	K504	0.30	0.55	106.88

(Continued)

Table 21 (continued)

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S206	0.50	0.45	125.60	K314	0.30	0.60	107.33	K505	0.30	0.55	107.03
S207	0.50	0.45	125.60	K315	0.30	0.60	107.33	K506	0.30	0.55	106.88
S208	0.50	0.60	168.59	K316	0.30	0.60	107.33	K507	0.30	0.55	108.61
S209	0.50	0.45	125.60	K317	0.30	0.60	107.67	K508	0.30	0.55	107.17
S210	0.50	0.30	87.39	S401	0.50	0.30	107.67	K509	0.30	0.55	108.61
S211	0.50	0.55	145.49	S102	0.50	0.55	77.83	K510	0.30	0.45	111.71
S212	0.50	0.30	87.39	S403	0.50	0.30	127.96	K511	0.30	0.45	111.71
K201	0.30	0.55	107.39	S404	0.50	0.45	77.83	K512	0.30	0.45	111.18
K202	0.30	0.55	107.08	S405	0.50	0.60	111.25	K513	0.30	0.45	111.18
K203	0.30	0.55	107.39	S406	0.50	0.45	139.89	K514	0.30	0.45	111.18
K204	0.30	0.55	106.95	S407	0.50	0.45	111.25	K515	0.30	0.45	111.18
K205	0.30	0.55	107.03	S408	0.50	0.60	111.25	K516	0.30	0.45	111.71
K206	0.30	0.55	106.95	S409	0.50	0.45	139.89	K517	0.30	0.45	111.71
K207	0.30	0.55	107.39	S410	0.50	0.30	111.25				
K208	0.30	0.55	116.08	S411	0.50	0.55	77.83				
				Total Cost							**17,042.4$**

In RC frame systems increasing the number of stories and applied loads substantially increases both axial loads and moment effects on columns. When compared with previous examples column sections are subjected to higher axial and moment effects and consequently designed with larger dimensions.

For beams, all studies show that increasing beam height to utilize the effective depth reduces reinforcement requirements. This approach increases the moment arm capacity, effectively resists applied moments, and provides more economical solutions.

The RC frame system with 145 elements undergoes static analysis in SAP2000 using optimal cross-section dimensions obtained through metaheuristic algorithm-based design. The MATLAB and SAP2000 results are compared to verify the analysis accuracy. Table 22 presents displacement values from both programs for four structural elements selected as one column and one beam from each floor. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 22: Comparison of displacement values obtained from MATLAB and SAP2000

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S101(i)	0	0	0	0	0	0
MATLAB		0	0	0	0	0	0
SAP2000	S101(j)	−0.000035	−0.000026	−0.00310	−0.00076	0.001557	0.000003007
MATLAB		−0.000036	−0.000026	−0.00310	0.00071	−0.001476	−0.000003007
SAP2000	K102(i)	−0.0000098	−0.000030	−0.00346	−0.00059	−0.000141	0.000000219
MATLAB		−0.0000095	−0.000029	−0.00346	0.00056	0.000137	−0.000000267
SAP2000	K102(j)	0.0000098	−0.000030	−0.00346	−0.00059	0.000141	−0.000000219
MATLAB		0.0000095	−0.000029	−0.00346	0.00056	−0.000137	0.000000267
SAP2000	S206(i)	−0.000019	0	−0.00351	0	−0.000014	0
MATLAB		−0.000019	0	−0.00351	0	0.000016	0

(Continued)

Table 22 (continued)

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S206(j)	−0.0000025	0	−0.00631	0	0.000021	0
MATLAB		−0.0000027	0	−0.00631	0	−0.000020	0
SAP2000	K203(i)	0.0000015	0.0000031	−0.00624	−0.00050	0.000078	−0.000000054
MATLAB		0.0000016	0.0000032	−0.00624	0.00046	−0.000073	0.000000069
SAP2000	K203(j)	0.00000091	0.0000037	−0.00559	−0.00063	−0.00140	−0.00000052
MATLAB		0.00000113	0.0000039	−0.00559	0.00059	0.00140	0.00000062
SAP2000	S209(i)	0.0000015	−0.0000031	−0.00624	0.00050	0.000078	0.000000055
MATLAB		0.0000016	−0.0000032	−0.00624	−0.00046	−0.000073	−0.000000069
SAP2000	S209(j)	0.0000015	−0.0000045	−0.00832	0.00053	0.000083	0.00000015
MATLAB		0.0000016	−0.0000050	−0.00832	−0.00049	−0.000080	−0.00000018
SAP2000	K311(i)	−0.0000015	0.0000046	−0.00745	−0.00065	0.00150	0.00000060
MATLAB		−0.0000016	0.0000050	−0.00745	−0.00061	−0.00142	−0.00000064
SAP2000	K311(j)	−0.0000022	0	−0.00710	0	0.000487	0
MATLAB		−0.0000022	0	−0.00711	0	−0.000468	0
SAP2000	S405(i)	−0.0000026	0	−0.00842	0	0.000017	0
MATLAB		−0.0000028	0	−0.00873	0	−0.000013	0
SAP2000	S405(j)	−0.000016	0	−0.00981	0	0.000056	0
MATLAB		−0.000016	0	−0.00981	0	−0.000056	0
SAP2000	K415(i)	0.0000014	−0.000043	−0.00971	−0.000402	0.000028	−0.00000055
MATLAB		0.0000011	−0.000044	−0.00971	0.000360	−0.000021	0.00000060
SAP2000	K415(j)	0.000016	0	−0.00981	0	−0.000056	0
MATLAB		0.000016	0	−0.00981	0	0.000056	0
SAP2000	S511(i)	0.000056	0	−0.00828	0	−0.000418	0
MATLAB		0.000056	0	−0.00829	0	0.000394	0
SAP2000	S511(j)	−0.000204	0	−0.00887	0	−0.001092	0
MATLAB		−0.000212	0	−0.00888	0	0.001049	0
SAP2000	K508(i)	0.000024	−0.00014	−0.01041	0.001359	−0.000226	−0.000000259
MATLAB		0.000024	−0.00014	−0.01041	−0.001302	0.000229	0.000000144
SAP2000	K508(j)	−0.000024	−0.00014	−0.01041	0.001359	0.000226	0.000000259
MATLAB		−0.000024	−0.00014	−0.01041	−0.001302	−0.000229	−0.000000144

The evaluation of displacement results obtained from MATLAB and SAP2000 analyses using the Mean Absolute Error (MAE) method is presented in Table 23. A comparison of values obtained from the code developed based on the Mean Absolute Error (MAE) method and the SAP2000 software revealed that the maximum difference in displacement values was 0.031 cm, while the maximum difference in rotation values was 0.00005 radians. The differences between the results presented in Table 23 have been determined to originate from the rounding errors used by the programs in their calculations.

Table 23: MAE comparison of displacement values: MATLAB vs. SAP2000

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000 MATLAB	S101(i)	0	0	0	0	0	0
SAP2000 MATLAB	S101(j)	0.000001	0	0	0.00005	0.000081	0
SAP2000 MATLAB	K102(i)	0.0000003	0.000001	0	0.00003	0.000005	0.00000005
SAP2000 MATLAB	K102(j)	0.0000003	0.000001	0	0.00003	0.000004	0.00000005
SAP2000 MATLAB	S206(i)	0	0	0	0	0.000002	0
SAP2000 MATLAB	S206(j)	0.0000002	0	0	0	0.000001	0
SAP2000 MATLAB	K203(i)	0.0000001	0.0000001	0	0.00004	0.000005	0.000000005
SAP2000 MATLAB	K203(j)	0.0000002	0.0000000.2	0	0.00004	0	0.0000001
SAP2000 MATLAB	S209(i)	0.0000001	0.0000001	0	0.00004	0.000005	0.000000014
SAP2000 MATLAB	S209(j)	0.0000001	0.0000005	0	0.00004	0.000003	0.00000004
SAP2000 MATLAB	K311(i)	0.0000001	0.0000004	0	0.00004	0.00008	0.00000004
SAP2000 MATLAB	K311(j)	0	0	0.00001	0	0.000019	0
SAP2000 MATLAB	S405(i)	0.0000002	0	0.00031	0	0.000004	0
SAP2000 MATLAB	S405(j)	0	0	0	0	0	0
SAP2000 MATLAB	K415(i)	0.0000003	0.000001	0	0.00004	0.000007	0.00000005
SAP2000 MATLAB	K415(j)	0	0	0	0	0	0
SAP2000 MATLAB	S511(i)	0	0	0.00001	0	0.000024	0
SAP2000 MATLAB	S511(j)	0.000008	0	0.00001	0	0.000043	0
SAP2000 MATLAB	K508(i)	0	0	0	0.00005	0.000003	0.000000115
SAP2000 MATLAB	K508(j)	0	0	0	0.00005	0.000003	0.000000115

3.4 Example 4

This section addresses optimizing a three-story RC frame system with two spans along the z-axis and three spans along the x-axis. The structure contains 36 columns and 51 beams, totaling 87 structural members. The design features asymmetric span lengths of 3 m along the z-axis and 4, 3, and 5 m along the x-axis, with uniform story heights of 3 m. Fig. 34 illustrates view of the frame system, loading conditions, and cross-section along the x-axis. The structure carries a 100 kN/m dead load and 50 kN/m live load, with load combinations using only vertical loads (1.4G + 1.6Q) applied.

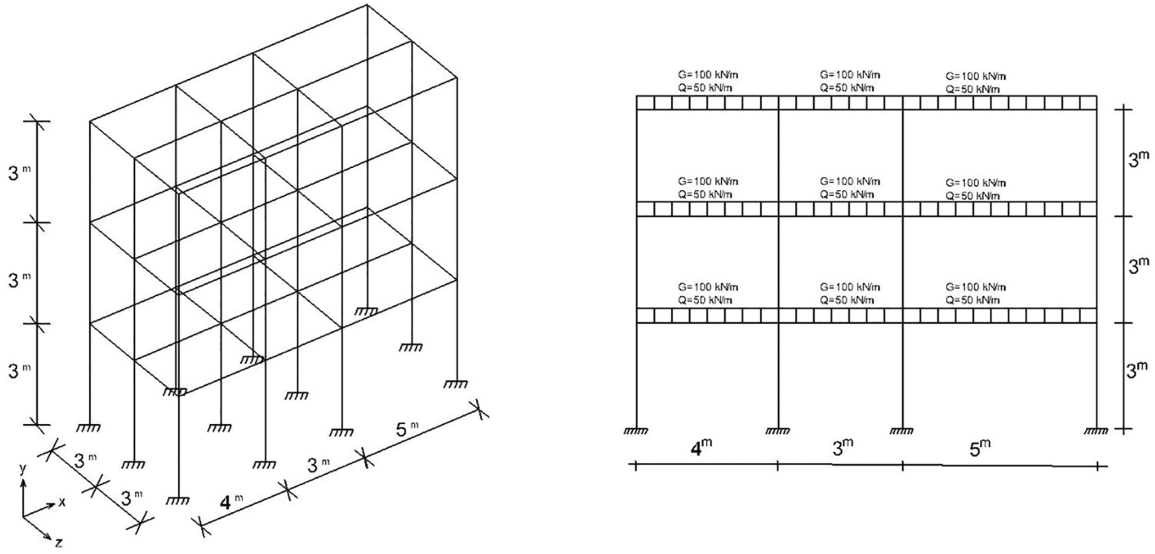

Figure 34: Load distribution and cross-section view of the 3-storey 3-span structure

Fig. 35 displays the floor plan showing column and beam layout for the RC frame system presented in Fig. 34. The design maintains constant cross-sectional areas for columns throughout the building height while allowing variable reinforcement quantities along columns. For beams, the design enforces uniform cross-sectional dimensions for all beams along the same axis to ensure construction efficiency.

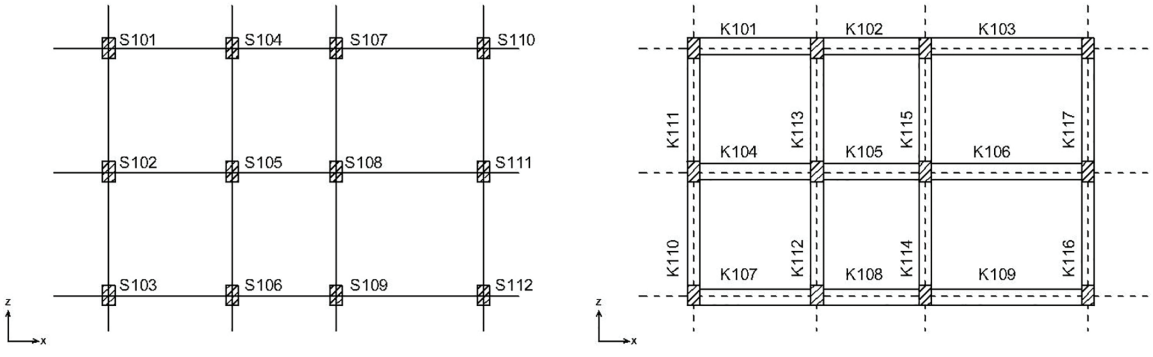

Figure 35: Column and beam layout plan of the 3-storey 3-span structure

Previous sections analyzed two different frame systems with 42 and 58 elements. A population size of 100 and 2000 iterations are utilized, with implemented code tracking and recording the exact

iteration step where optimization completes. Each analysis runs 10 repetitions, and Table 24 presents the optimal cost achieved for the structure and the iteration number reaching the optimum result.

Table 24: Cost optimization results obtained from the TLBO algorithm for the 87 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	7475.5	7475.5	7475.5	7475.5	7475.5	7486.8	7475.5	7475.5	7475.5	7486.8	7475.5	7486.8	5.05
Optimization completion step	135	144	166	128	116	165	174	113	285	142	113	165	–

The TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 36 presents the optimal cost distribution obtained from these analyses.

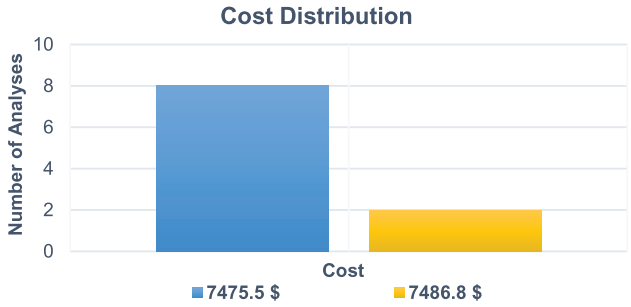

Figure 36: Distribution of analysis results from the TLBO algorithm

The convergence behavior of Analysis 8 which achieved the fastest optimization in the TLBO algorithm is shown in Fig. 37. It is seen that the algorithm reaches the optimum result at the 113th iteration.

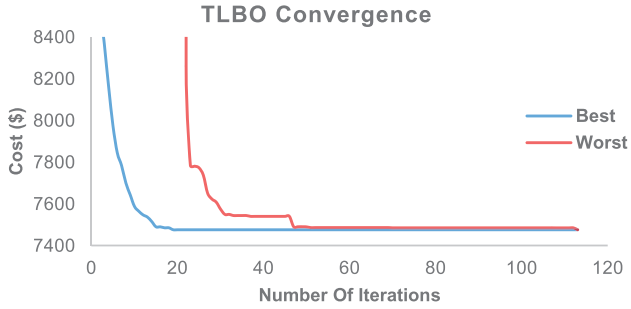

Figure 37: Convergence graph of the TLBO algorithm

Table 25 displays the optimized section dimensions and corresponding cost values for each structural member of the 87-element RC space frame system obtained from the computational analysis.

Table 25: Section dimensions and optimum cost values of the 87-element RC system

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S101	0.30	0.30	60.34	K204	0.30	0.60	103.57
S102	0.50	0.30	82.61	K205	0.30	0.60	73.15
S103	0.30	0.30	60.34	K206	0.30	0.60	159.18
S104	0.40	0.30	88.69	K207	0.30	0.60	104.41
S105	0.45	0.40	105.62	K208	0.30	0.60	75.57
S106	0.40	0.30	88.69	K209	0.30	0.60	160.25
S107	0.45	0.35	92.05	K210	0.30	0.45	61.93
S108	0.50	0.40	117.64	K211	0.30	0.45	61.93
S109	0.45	0.35	92.05	K212	0.30	0.45	61.73
S110	0.30	0.30	63.21	K213	0.30	0.45	61.73
S111	0.30	0.45	100.00	K214	0.30	0.45	61.55
S112	0.30	0.30	63.21	K215	0.30	0.45	61.55
K101	0.30	0.60	105.57	K216	0.30	0.45	61.79
K102	0.30	0.60	76.80	K217	0.30	0.45	61.79
K103	0.30	0.60	160.55	S301	0.30	0.30	66.08
K104	0.30	0.60	104.53	S302	0.50	0.30	68.26
K105	0.30	0.60	74.61	S303	0.30	0.30	66.08
K106	0.30	0.60	159.23	S304	0.40	0.30	77.21
K107	0.30	0.60	105.57	S305	0.45	0.40	82.67
K108	0.30	0.60	76.80	S306	0.40	0.30	77.21
K109	0.30	0.60	160.55	S307	0.45	0.35	97.07
K110	0.30	0.50	64.79	S308	0.50	0.40	92.14
K111	0.30	0.50	64.79	S309	0.45	0.35	97.07
K112	0.30	0.50	63.52	S310	0.30	0.30	77.55
K113	0.30	0.50	63.52	S311	0.30	0.45	112.91
K114	0.30	0.50	62.47	S312	0.30	0.30	77.55
K115	0.30	0.50	62.47	K301	0.30	0.60	107.52
K116	0.30	0.50	64.46	K302	0.30	0.60	79.08
K117	0.30	0.50	64.46	K303	0.30	0.60	163.35
S201	0.30	0.30	54.60	K304	0.30	0.60	106.45
S202	0.50	0.30	68.26	K305	0.30	0.60	76.95
S203	0.30	0.30	54.60	K306	0.30	0.60	159.98
S204	0.40	0.30	61.91	K307	0.30	0.60	107.52
S205	0.45	0.40	88.41	K308	0.30	0.60	79.08
S206	0.40	0.30	61.91	K309	0.30	0.60	163.35
S207	0.45	0.35	76.99	K310	0.30	0.40	62.98
S208	0.50	0.40	98.52	K311	0.30	0.40	62.98
S209	0.45	0.35	76.99	K312	0.30	0.40	62.31
S210	0.30	0.30	71.81	K313	0.30	0.40	62.31
S211	0.30	0.45	95.69	K314	0.30	0.40	61.96
S212	0.30	0.30	71.81	K315	0.30	0.40	61.96

(Continued)

Table 25 (continued)

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
K201	0.30	0.60	104.41	K316	0.30	0.40	62.45
K202	0.30	0.60	75.57	K317	0.30	0.40	62.45
K203	0.30	0.60	160.25				
			Total Cost				**7475.5$**

The RC frame system with 87 elements undergoes static analysis in SAP2000 using optimal cross-section dimensions obtained through a metaheuristic algorithm-based design. The MATLAB and SAP2000 results are compared to verify the accuracy of the analysis. Table 26 presents displacement values from both programs for four structural elements selected as one column and one beam from each floor. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 26: Comparison of displacement values obtained from MATLAB and SAP2000

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S101(i)	0	0	0	0	0	0
MATLAB		0	0	0	0	0	0
SAP2000	S101(j)	−0.000119	−0.0000091	−0.00265	−0.001033	0.00194	−0.000012
MATLAB		−0.000132	−0.0000089	0.00265	0.000097	−0.00189	−0.000012
SAP2000	K105(i)	−0.000037	0	−0.00293	0	−0.00061	0
MATLAB		−0.000051	0	−0.00294	0	0.00062	0
SAP2000	K105(j)	−0.000029	0	−0.00289	0	0.00091	0
MATLAB		−0.000043	0	−0.00289	0	−0.00090	0
SAP2000	S209(i)	−0.000098	0.000014	−0.00287	0.00055	0.00124	0.00000614
MATLAB		−0.000111	0.000014	−0.00287	−0.00054	−0.00122	−0.00000580
SAP2000	S209(j)	−0.000121	0.00002	−0.00480	0.00042	0.00104	−0.00000483
MATLAB		−0.000152	0.00002	−0.00480	−0.00041	−0.00100	−0.00000574
SAP2000	K209(i)	−0.000121	0.00002	−0.00480	0.00042	0.00104	−0.00000483
MATLAB		−0.000152	0.00002	−0.00480	−0.00041	−0.00101	−0.00000574
SAP2000	K209(j)	−0.000112	0.0000093	−0.00501	0.00115	−0.00273	−0.00000791
MATLAB		−0.000143	0.0000091	−0.00501	−0.00109	0.00268	0.00000104
SAP2000	S303(i)	−0.000126	0.0000091	−0.00443	0.00107	0.00182	−0.00000933
MATLAB		−0.000157	0.0000089	−0.00443	−0.00101	−0.00178	0.00001097
SAP2000	S303(j)	0.000021	−0.000032	−0.00530	0.00198	0.00248	−0.000073
MATLAB		−0.000024	−0.000031	−0.00530	−0.00188	−0.00244	0.000076
SAP2000	K306(i)	−0.000416	0	−0.00581	0	0.00131	0
MATLAB		−0.000477	0	−0.00581	0	−0.00128	0
SAP2000	K306(j)	−0.000558	0	−0.00648	0	−0.00267	0
MATLAB		−0.000623	0	−0.00648	0	−0.00260	0

The evaluation of displacement results obtained from MATLAB and SAP2000 analyses using the Mean Absolute Error (MAE) method is presented in Table 27. A comparison of values obtained from the code developed based on the Mean Absolute Error (MAE) method and the SAP2000 software revealed that the maximum difference in displacement values was 0.0065 cm, while the maximum difference in rotation values was 0.00093 radians. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 27: MAE comparison of displacement values: MATLAB vs. SAP2000

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000 MATLAB	S101(i)	0	0	0	0	0	0
SAP2000 MATLAB	S101(j)	0.000013	0.0000003	0	0.00093	0.00005	0
SAP2000 MATLAB	K105(i)	0.000014	0	0.00001	0	0.00001	0
SAP2000 MATLAB	K105(j)	0.000014	0	0	0	0.00001	0
SAP2000 MATLAB	S209(i)	0.000013	0	0	0.00001	0.00002	0.00000034
SAP2000 MATLAB	S209(j)	0.000031	0	0	0.00001	0.00004	0.0000009
SAP2000 MATLAB	K209(i)	0.000031	0	0	0.00001	0.00003	0.0000009
SAP2000 MATLAB	K209(j)	0.000031	0.0000002	0	0.00006	0.00005	0.0000068
SAP2000 MATLAB	S303(i)	0.000031	0.0000003	0	0.00006	0.00004	0.0000016
SAP2000 MATLAB	S303(j)	0.000001	0.000001	0	0.0001	0.00004	0.000003
SAP2000 MATLAB	K306(i)	0.000061	0	0	0	0.00003	0
SAP2000 MATLAB	K306(j)	0.000065	0	0	0	0.00007	0

3.5 Example 5

This section designs the RC space frame system differently from the rectangular forms in the previous sections. The building floor plan has an L-shape. The structure has single spans on the first x-axis and three spans on other x-axis, while having single spans on the first two z-axes and two spans

on the last z-axis. The two-story RC frame system with two spans in the z-direction and three spans in the x-direction contains 20 columns and 26 beams, totaling 46 structural members. The design features 4-m span lengths and 3-m story heights. Fig. 38 shows view of the frame system, loading conditions, and cross-section along the x-axis. The structure carries a 100 kN/m dead load and 50 kN/m live load using only vertical load combinations (1.4G + 1.6Q).

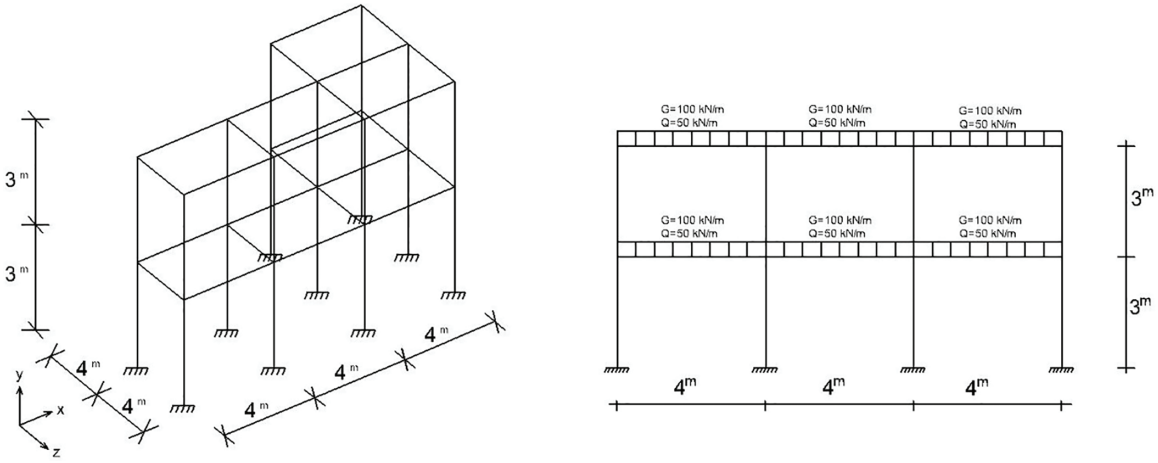

Figure 38: Load distribution and cross-section view of the 46-member structure

Fig. 39 displays the floor plan showing column and beam layout for the RC frame system presented in Fig. 38. The design maintains constant cross-sectional areas for columns throughout the building height while allowing variable reinforcement quantities along columns. For beams, the design enforces equal cross-sectional dimensions for all beams along the same axis to ensure construction efficiency.

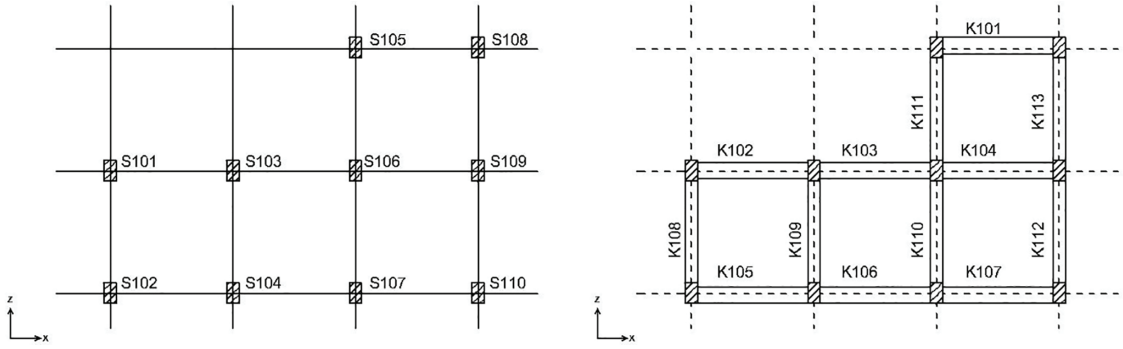

Figure 39: Column and beam layout plan of the 46-member structure

A population size of 100 and 2000 iterations are utilized, with implemented code tracking and recording the exact iteration step where optimization completes. Each analysis runs 10 repetitions and Table 28 presents the optimal cost achieved for the structure and the iteration number reaching the optimum result.

Table 28: Cost optimization results obtained from the TLBO algorithm for the 46 element structure

Analysis number	1	2	3	4	5	6	7	8	9	10	Best solution	Worst solution	Standard deviation
Cost (USD)	4421.9	4421.9	4425.4	4421.9	4421.9	4421.9	4425.4	4421.9	4421.9	4421.9	4421.9	4425.4	1.57
Optimization completion step	77	89	96	87	104	154	97	76	91	70	76	97	–

The TLBO algorithm achieves the global optimum in 8 out of 10 conducted analyses. Fig. 40 presents the optimal cost distribution obtained from these analyses.

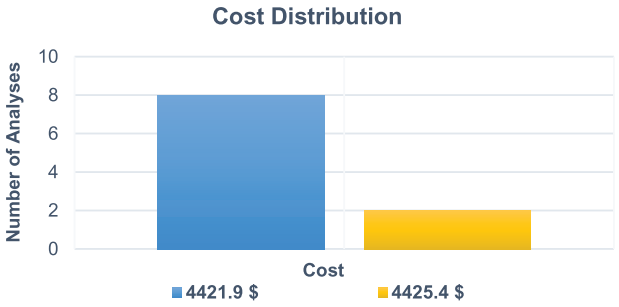

Figure 40: Distribution of analysis results from the TLBO algorithm

The convergence behavior of Analysis 10 which achieved the fastest optimization in the TLBO algorithm is shown in Fig. 41. It is seen that the algorithm reaches the optimum result at the 70th iteration.

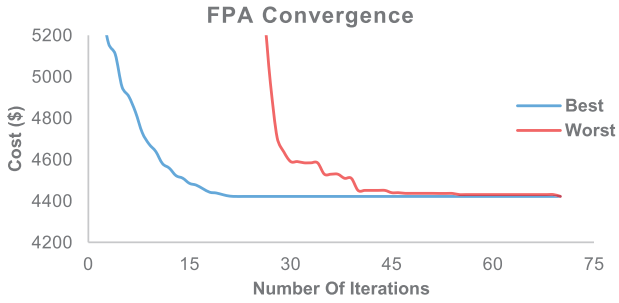

Figure 41: Convergence graph of the TLBO algorithm

Table 29 displays the optimized section dimensions and corresponding cost values for each structural member of the 46-element RC space frame system obtained from the computational analysis.

Table 29: Section dimensions and optimum cost values of the 46-element RC system

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S101	0.30	0.30	2431.0	S201	0.30	0.30	3424.2

(Continued)

Table 29 (continued)

Element name	b_w (m)	h (m)	Cost (₺)	Element name	b_w (m)	h (m)	Cost ($)
S102	0.30	0.30	2431.0	S202	0.30	0.30	3424.2
S103	0.35	0.30	2976.3	S203	0.35	0.30	3748.8
S104	0.35	0.30	2976.3	S204	0.35	0.30	3877.5
S105	0.30	0.30	2210.3	S205	0.30	0.30	2872.5
S106	0.45	0.35	3540.4	S206	0.45	0.35	2767.9
S107	0.35	0.30	2590.0	S207	0.35	0.30	3105.0
S108	0.30	0.30	2100.0	S208	0.30	0.30	3203.5
S109	0.45	0.30	2852.8	S209	0.45	0.30	2521.8
S110	0.30	0.30	2100.0	S210	0.30	0.30	3203.5
K101	0.30	0.55	4172.9	K201	0.30	0.55	4232.0
K102	0.30	0.55	4209.8	K202	0.30	0.60	4382.4
K103	0.30	0.55	4180.8	K203	0.30	0.60	4272.3
K104	0.30	0.55	4197.8	K204	0.30	0.60	4363.4
K105	0.30	0.55	4202.8	K205	0.30	0.50	4262.0
K106	0.30	0.55	4180.0	K206	0.30	0.50	4226.4
K107	0.30	0.55	4200.7	K207	0.30	0.50	4270.1
K108	0.35	0.60	4710.4	K208	0.30	0.60	4421.7
K109	0.30	0.60	4178.7	K209	0.30	0.60	4280.2
K110	0.30	0.60	4320.7	K210	0.30	0.60	4432.7
K111	0.30	0.60	4290.1	K211	0.30	0.60	4404.3
K112	0.30	0.60	4326.4	K212	0.30	0.50	4345.1
K113	0.30	0.60	4318.6	K213	0.30	0.50	4337.4
			Total Cost				**4421.9$**

The 46-element RC frame system, optimized using metaheuristic algorithms, was analyzed in SAP2000 with the obtained cross-sectional dimensions. The results from MATLAB and SAP2000 were compared to verify the accuracy of the analysis. Table 30 presents the displacement values from both programs for four structural elements (one column and one beam selected from each floor). The coordinate system notation in the comparison table follows SAP2000 conventions, where Z denotes the vertical building height direction. Due to the absence of horizontal loading and the symmetric, distributed nature of the applied loads, the displacement values in the X and Y directions are significantly lower than in other directions. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 30: Comparison of displacement values obtained from MATLAB and SAP2000

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
SAP2000	S101(i)	0	0	0	0	0	0

(Continued)

Table 30 (continued)

		X (m)	Y (m)	Z (m)	θx (rad)	θy (rad)	θz (rad)
MATLAB		0	0	0	0	0	0
SAP2000	S101(j)	−0.000098	−0.00002	−0.00213	−0.00204	0.00213	−0.00000511
MATLAB		−0.000094	−0.00002	−0.00213	0.00200	−0.00208	0.00000518
SAP2000	K103(i)	−0.000073	−0.00004	−0.00305	−0.00171	−0.00033	−0.00000475
MATLAB		−0.000069	−0.00004	−0.00305	0.00165	0.00034	0.00000344
SAP2000	K103(j)	−0.000053	−0.00005	−0.00291	−0.000043	0.00027	−0.00000301
MATLAB		−0.000051	−0.00005	−0.00291	0.000058	0.00032	−0.00000363
SAP2000	S207(i)	−0.000025	−0.00002	−0.00289	0.00134	0.00026	−0.00000294
MATLAB		−0.000017	−0.00002	−0.00289	−0.00131	−0.00027	−0.00000341
SAP2000	S207(j)	−0.000121	−0.00015	−0.00434	0.00176	0.00048	−0.00000541
MATLAB		−0.000103	−0.00012	−0.00434	−0.00174	−0.00049	−0.00000139
SAP2000	K212(i)	−0.000239	−0.00010	−0.00363	0.000019	−0.00208	−0.00000433
MATLAB		−0.000205	−0.00009	−0.00363	−0.000007	0.00208	−0.00000126
SAP2000	K212(j)	−0.000182	−0.00015	−0.00292	0.00287	−0.00322	−0.00000671
MATLAB		−0.000163	−0.00014	−0.00292	−0.00277	0.00312	−0.00000694

The evaluation of displacement results obtained from MATLAB and SAP2000 analyses using the Mean Absolute Error (MAE) method is presented in Table 31. A comparison of values obtained from the code developed based on the Mean Absolute Error (MAE) method and the SAP2000 software revealed that the maximum difference in displacement values was 0.003 cm, while the maximum difference in rotation values was 0.0001 radians. The differences between the results presented in Table 31 have been determined to originate from the rounding errors used by the programs in their calculations. It was observed that the displacements and rotations generated using our developed code were consistent with the values obtained from the software package, and the differences between the results were determined to be within acceptable limits from a civil engineering perspective.

Table 31: MAE comparison of displacement values: MATLAB vs. SAP2000

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000	S101(i)	0	0	0	0	0	0
MATLAB							
SAP2000	S101(j)	0.000004	0	0	0.00004	0.00005	0.00000007
MATLAB							
SAP2000	K103(i)	0.000004	0	0	0.00006	0.00001	0.0000013
MATLAB							
SAP2000	K103(j)	0.000002	0	0	0.000015	0.00005	0.0000006
MATLAB							
SAP2000	S207(i)	0.000008	0	0	0.00003	0.00001	0.00000047
MATLAB							
SAP2000	S207(j)	0.000018	0.00003	0	0.00002	0.00001	0.0000040
MATLAB							
SAP2000	K212(i)	0.000034	0.00001	0	0.000012	0	0.00000307
MATLAB							

(Continued)

Table 31 (continued)

		X (m) MAE	Y (m) MAE	Z (m) MAE	θx (rad) MAE	θy (rad) MAE	θz (rad) MAE
SAP2000	K212(j)	0.000019	0.00001	0	0.0001	0.0001	0.00000023
MATLAB							

4 Conclusions

This study examines five different RC space frame systems subjected to vertical loads. The first two case studies utilize five optimization algorithms (Jaya algorithm, Teaching-Learning-Based Optimization (TLBO) algorithm, Flower Pollination Algorithm (FPA), JA/TLBO hybrid algorithm, and JA/FPA hybrid algorithm). The TLBO algorithm, which demonstrated the best performance among the optimization algorithms, was subsequently used to analyze the remaining three structural systems.

Based on the first two analyses, the Teaching-Learning-Based Optimization (TLBO) algorithm emerges as the most effective method for optimal cost design of space RC frame systems. The third case study examines a five-story RC structure with 145 elements, maintaining symmetry like the second example. TLBO-based cost optimization yields a total structural cost of $17,042.4. Due to high axial loads, column dimensions increase significantly. All algorithms in this study accommodate axial loads by enlarging column sections, while beam design optimizes cost by increasing only beam height to reduce reinforcement area for moment resistance. This approach achieves optimal structural costs through:

Symmetrically planned reinforced concrete structures are examined in the first three examples. The fourth case performs cost optimization on a three-story RC frame system with 87 elements, featuring two spans in one direction and three unequal spans in the other direction. The TLBO algorithm calculates the frame system's construction cost as $7475.5. Increasing span lengths results in greater beam depths. The algorithm provides an economical solution by increasing beam depth to enhance moment capacity while using less reinforcement.

The final case study in this research examines a completely asymmetric structure. The L-shaped floor plan features span lengths throughout the 46-element RC frame system. The TLBO algorithm calculates the optimized construction cost as $4421.9. The TLBO algorithm achieves optimal results for this asymmetric frame system in 8 out of 10 analyses, demonstrating its effectiveness for static analysis and cost optimization of space RC frame systems.

Despite the increase in the number of variables, the optimum solution can be reached by the algorithm. Although the convergence step is larger in asymmetric structures compared to symmetric ones, it has been observed that the TLBO algorithm is effective in achieving the optimum solution for both types of structures.

Five different RC frame systems subjected to vertical loads are designed and cost-optimized. While all algorithms achieved optimal results, the Teaching-Learning-Based Optimization (TLBO) algorithm proved most suitable and effective for cost optimization of RC frame systems when considering convergence speed, stability, and standard deviation values. Structural static analysis was performed in MATLAB using the matrix displacement method, enabling optimal cost design based on derived section forces. The optimized cross-sections were verified through additional analysis in SAP2000. To validate the accuracy of the analysis, all RC structure examples were analyzed in both programs, comparing displacement values in columns and beams. The close agreement between

MATLAB and SAP2000 results for horizontal/vertical displacements and rotations at member ends confirms the reliability of MATLAB's static analysis.

The Jaya algorithm encountered local optima trapping problems when working with discrete variables. This trapping in local optima negatively affected the algorithm's performance and limited solution accuracy. The TLBO (Teaching-Learning-Based Optimization) and FPA (Flower Pollination Algorithm) were also employed to obtain better results. Test results demonstrated that the TLBO algorithm yielded the best outcomes.

Considering that the Jaya algorithm could be successfully applied to other structural engineering problems, and to overcome its local optima trapping issue, two hybrid algorithms (JA/TLBO and JA/FPA) were developed by separately combining it with TLBO (Teaching-Learning-Based Optimization) and FPA (Flower Pollination Algorithm). The hybrid models integrate the strengths of each algorithm to enhance Jaya's performance. While the hybrid algorithms achieved optimal results as consistently as TLBO, they did not match TLBO's convergence speed. In this context, the TLBO algorithm emerged as the most efficient method delivering the best results.

Using metaheuristic algorithms in RC frame system design enables structures to effectively carry applied loads while achieving optimal dimensioning and reinforcement under the most economical conditions within structural design requirements.

This study excluded slabs, foundations, and formwork from the cost estimation in order to focus on the primary superstructure elements. However, it is important to recognize that this omission leads to an underestimation of total construction cost. The type of slab system (e.g., flat slab, waffle slab, ribbed slab) is particularly influential in cost optimization, as slab design affects both material consumption and construction complexity. Furthermore, formwork is a more significant cost factor for slabs than for beams and columns, due to the large surface areas and extensive temporary supports required.

Foundations also represent a major cost component, with the selected system (e.g., shallow footings, raft foundations, or deep pile foundations) heavily dependent on soil conditions and loading requirements. Since foundations can constitute a substantial portion of total structural costs, their exclusion limits the applicability of the current estimates to real-world budgeting.

In future studies, the developed method is intended to be extended by including dynamic loads such as wind and seismic effects, as well as by incorporating other significant cost factors such as formwork, labor, overhead, slab, and foundation costs to provide the total project cost.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Sinan Melih Nigdeli, Gebrail Bekdaş; data collection: Yasin Duysak; analysis and interpretation of results: Yasin Duysak, Sinan Melih Nigdeli, Gebrail Bekdaş; draft manuscript preparation: Yasin Duysak. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Rao RV. Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comput. 2016;7(1):19–34. doi:10.5267/j.ijiec.2015.8.004.
2. Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des. 2011;43(3):303–15. doi:10.1016/j.cad.2010.12.015.
3. Goldberg DE, Deb K. A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of genetic algorithms. Amsterdam, The Netherland: Elsevier; 1991. p. 69–93. doi:10.1016/b978-0-08-050684-5.50008-2.
4. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59. doi:10.1023/A:1008202821328.
5. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag. 2006;1(4):28–39. doi:10.1109/ci-m.2006.248054.
6. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95—International Conference on Neural Networks; 1995 Nov 27–Dec 1; Perth, WA, Australia. p. 1942–8. doi:10.1109/ICNN.1995.488968.
7. Yang XS, Deb S. Cuckoo search via lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC); 2009 Dec 9–11; Coimbatore, India. p. 210–4. doi:10.1109/NABIC.2009.5393690.
8. Erol OK, Eksin I. A new optimization method: big Bang-Big Crunch. Adv Eng Softw. 2006;37(2):106–11. doi:10.1016/j.advengsoft.2005.04.005.
9. Yang XS. Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation. Berlin/Heidelberg, Germany: Springer; 2012. p. 240–9. doi:10.1007/978-3-642-32894-7_27.
10. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation. 2001;76(2):60–8. doi:10.1177/003754970107600201.
11. Yang XS. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Berlin/Heidelberg, Germany: Springer; 2010. p. 65–74. doi:10.1007/978-3-642-12538-6_6.
12. Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simulated annealing. Science. 1983;220(4598):671–80. doi:10.1126/science.220.4598.671.
13. Yang XS. Nature-inspired metaheuristic algorithms. 2nd ed. Frome, UK: Luniver Press; 2010.
14. Simon D. Biogeography-based optimization. IEEE Trans Evol Comput. 2008;12(6):702–13. doi:10.1109/TEVC.2008.919004.
15. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95:51–67. doi:10.1016/j.advengsoft.2016.01.008.
16. Mohammadi-Balani A, Dehghan Nayeri M, Azar A, Taghizadeh-Yazdi M. Golden eagle optimizer: a nature-inspired metaheuristic algorithm. Comput Ind Eng. 2021;152:107050. doi:10.1016/j.cie.2020.107050.
17. Rafiq MY, Southcombe C. Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. Comput Struct. 1998;69(4):443–57. doi:10.1016/S0045-7949(98)00108-4.
18. Bekdas G, Nigdeli SM. The optimization of slender reinforced concrete columns. PAMM. 2014;14(1):183–4. doi:10.1002/pamm.201410079.
19. Bekdaş G, Nigdeli SM. Optimum design of reinforced concrete columns employing teaching-learning based optimization. Challenge J Struct Mech. 2016;2(4):216–9. doi:10.20528/cjsmec.2016.11.030.

20. Nigdeli SM, Bekdas G, Kim S, Geem ZW. A novel harmony search based optimization of reinforced concrete biaxially loaded columns. Struct Eng Mech. 2015;54(6):1097–109. doi:10.12989/sem.2015.54.6.1097.
21. Bekdas G, Nigdeli SM. Bat algorithm for optimization of reinforced concrete columns. PAMM. 2016;16(1):681–2. doi:10.1002/pamm.201610329.
22. de Medeiros GF, Kripka M. Optimization of reinforced concrete columns according to different environmental impact assessment parameters. Eng Struct. 2014;59:185–94. doi:10.1016/j.engstruct.2013.10.045.
23. Ozturk HT, Durmus A. Optimum cost design of RC columns using artificial bee colony algorithm. Struct Eng Mech. 2013;45(5):643–54. doi:10.12989/sem.2013.45.5.643.
24. Bekdaş G, Cakiroglu C, Kim S, Geem ZW. Optimization and predictive modeling of reinforced concrete circular columns. Materials. 2022;15(19):6624. doi:10.3390/ma15196624.
25. Guimarães SA, Klein D, Calenzani AFG, Alves ÉC. Optimum design of steel columns filled with concrete via genetic algorithm: environmental impact and cost analysis. REM Int Eng J. 2022;75(2):117–28. doi:10.1590/0370-44672021750034.
26. Preethi G, Arulraj PG. Optimal design of axially loaded RC columns. Bonfring Int J Ind Eng Manag Sci. 2016;6(3):78–81. doi:10.9756/bijiems.7345.
27. Rao HS, Babu BR. Optimized column design using genetic algorithm based neural networks. Indian J Eng Mat Scis. 2006;13(6):503–11. doi:10.1007/s00521-020-05605-z.
28. Medeiros GF, Kripka M. Modified harmony search and its application to cost minimization of RC columns. Adv Comput Des. 2017;2(1):1–13. doi:10.12989/acd.2017.2.1.001.
29. Boscardin JT, Yepes V, Kripka M. Optimization of reinforced concrete building frames with automated grouping of columns. Autom Constr. 2019;104:331–40. doi:10.1016/j.autcon.2019.04.024.
30. Duysak Y, Nigdeli SM, Bekdaş G. Optimum design of reinforced concrete beam sections with JAYA algorithm. Chall J Concr Res Lett. 2024;15(4):134. doi:10.20528/cjcrl.2024.04.003.
31. Duysak Y, Bekdaş G. Investigation of effect of flexural and torsional moment values on optimum RC beam design. Wseas Trans Appl Theor Mech. 2024;19:189–99. doi:10.37394/232011.2024.19.21.
32. Jahjouh MM, Arafa MH, Alqedra MA. Artificial Bee Colony (ABC) algorithm in the design optimization of RC continuous beams. Struct Multidiscip Optim. 2013;47(6):963–79. doi:10.1007/s00158-013-0884-y.
33. Lepš M, Šejnoha M. New approach to optimization of reinforced concrete beams. Comput Struct. 2003;81(18–19):1957–66. doi:10.1016/S0045-7949(03)00215-3.
34. Pierott R, Hammad AWA, Haddad A, Garcia S, Falcón G. A mathematical optimisation model for the design and detailing of reinforced concrete beams. Eng Struct. 2021;245(1):112861. doi:10.1016/j.engstruct.2021.112861.
35. Deliktaş B, Bikçe M, Coşkun H, Türker HT. Betonarme kirişlerin optimum tasariminda genetik algoritma parametrelerinin etkisinin belirlenmesi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi. 2009;2(2):125–32. doi:10.6088/ijaser.020100009.
36. T. YS, Mh NR. Optimum cost design of reinforced concrete continuous beams using Genetic Algorithms. Int J Appl Sci Eng Res. 2013;2(1):79–92. doi:10.6088/ijaser.020100009.
37. Aydın Y, Bekdaş G, Nigdeli SM. Reinforced concrete beam optimization via flower pollination algorithm by changing switch probability parameter. In: Intelligent computing and optimization. Cham, Switzerland: Springer Nature; 2023. p. 66–74. doi:10.1007/978-3-031-50330-6_7.
38. Bekdaş G, Nigdeli SM. Rassal Arama Tekniği İle Betonarme Kirişlerin Farklı Beton Dayanımları İçin Optimizasyonu. In: XVIII. Ulusal Mekanik Kongresi; 2013 Aug 26–30; Manisa, Turkıye. p. 105–10.
39. Bekdaş G, Nigdeli SM. Optimization of t-shaped RC flexural members for different compressive strengths of concrete. Int J Mech. 2013;7(2):109–19. doi:10.12989/scs.2020.34.3.409.
40. Tormen AF, Pravia ZMC, Ramires FB, Kripka M. Optimization of steel-concrete composite beams considering cost and environmental impact. Steel Compos Struct Int J. 2020;34(3):409–21. doi:10.12989/scs.2020.34.3.409.

41. Tamrazyan A, Alekseytsev AV. Optimization of reinforced concrete beams under local mechanical and corrosive damage. Eng Optim. 2023;55(11):1905–22. doi:10.1080/0305215X.2022.2134356.
42. Alqedra M, Arafa M, Ismail M. Optimum cost of prestressed and reinforced concrete beams using genetic algorithms. J Artif Intell. 2010;4(1):76–88. doi:10.3923/jai.2011.76.88.
43. Fedghouche F, Tiliouine B. Minimum cost design of reinforced concrete T-beams at ultimate loads using Eurocode2. Eng Struct. 2012;42:43–50. doi:10.1016/j.engstruct.2012.04.008.
44. Luévanos Rojas A. Numerical experimentation for the optimal design of reinforced rectangular concrete beams for singly reinforced sections. Dyna. 2016;83(196):134–42. doi:10.15446/dyna.v83n196.48031.
45. Chutani S, Singh J. Design optimization of reinforced concrete beams. J Inst Eng Ind Ser A. 2017;98(4):429–35. doi:10.1007/s40030-017-0232-0.
46. Sharafi P, Hadi MNS, Teh LH. Geometric design optimization for dynamic response problems of continuous reinforced concrete beams. J Comput Civ Eng. 2014;28(2):202–9. doi:10.1061/(asce)cp.1943-5487.0000263.
47. Coello CC, Santos Hernández F, Farrera FA. Optimal design of reinforced concrete beams using genetic algorithms. Expert Syst Appl. 1997;12(1):101–8. doi:10.1016/S0957-4174(96)00084-X.
48. Govindaraj V, Ramasamy JV. Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms. Comput Struct. 2005;84(1–2):34–48. doi:10.1016/j.compstruc.2005.09.001.
49. Saini B, Sehgal VK, Gambhir ML. Genetically optimized artificial neural network based optimum design of singly and doubly reinforced concrete beams. Asian J Civil Eng. 2006;7(6):603–19. doi:10.1590/s1679-78252014000700007.
50. García-Segura T, Yepes V, Martí JV, Alcalá J. Optimization of concrete I-beams using a new hybrid glowworm swarm algorithm. Lat Am J Solids Struct. 2014;11(7):1190–205. doi:10.1590/s1679-78252014000700007.
51. Rahmanian I, Lucet Y, Tesfamariam S. Optimal design of reinforced concrete beams: a review. Comput Concr. 2014;13(4):457–82. doi:10.12989/cac.2014.13.4.457.
52. Kaveh A, Zakian P. Optimal seismic design of Reinforced Concrete shear wall-frame structures. KSCE J Civ Eng. 2014;18(7):2181–90. doi:10.1007/s12205-014-0640-x.
53. Kaveh A, Zakian P. Performance based optimal seismic design of RC shear walls incorporating soil-structure interaction using CSS algorithm. Int J Optimiz Civil Eng. 2012;2(3):383–405. doi:10.1007/bf01200046.
54. Fadaee MJ, Grierson DE. Design optimization of 3D reinforced concrete structures having shear walls. Eng Comput. 1998;14(2):139–45. doi:10.1007/BF01213587.
55. Zhang Y, Mueller C. Shear wall layout optimization for conceptual design of tall buildings. Eng Struct. 2017;140:225–40. doi:10.1016/j.engstruct.2017.02.059.
56. Atabay Ş. Cost optimization of three-dimensional beamless reinforced concrete shear-wall systems via genetic algorithm. Expert Syst Appl. 2009;36(2):3555–61. doi:10.1016/j.eswa.2008.02.004.
57. Cerè G, Rezgui Y, Zhao W, Petri I. Shear walls optimization in a reinforced concrete framed building for seismic risk reduction. J Build Eng. 2022;54(1):104620. doi:10.1016/j.jobe.2022.104620.
58. Titiksh A, Bhatt G. Optimum positioning of shear walls for minimizing the effects of lateral forces in multistorey-buildings. Arch Civ Eng. 2017;63(1):151–62. doi:10.1515/ace-2017-0010.
59. Camp CV, Pezeshk S, Hansson H. Flexural design of reinforced concrete frames using a genetic algorithm. J Struct Eng. 2003;129(1):105–15. doi:10.1061/(asce)0733-9445(2003)129:1(105).
60. Razmara Shooli A, Vosoughi AR, Banan MR. A mixed GA-PSO-based approach for performance-based design optimization of 2D reinforced concrete special moment-resisting frames. Appl Soft Comput. 2019;85:105843. doi:10.1016/j.asoc.2019.105843.
61. Zou XK, Chan CM, Li G, Wang Q. Multiobjective optimization for performance-based design of reinforced concrete frames. J Struct Eng. 2007;133(10):1462–74. doi:10.1061/(asce)0733-9445(2007)133:10(1462).

62. Fragiadakis M, Papadrakakis M. Performance-based optimum seismic design of reinforced concrete structures. Earthq Eng Struct Dyn. 2008;37(6):825–44. doi:10.1002/eqe.786.
63. Guerra A, Kiousis PD. Design optimization of reinforced concrete structures. Comput Concr. 2006;3(5):313–34. doi:10.12989/cac.2006.3.5.313.
64. Tapao A, Cheerarot R. Optimal parameters and performance of artificial bee colony algorithm for minimum cost design of reinforced concrete frames. Eng Struct. 2017;151:802–20. doi:10.1016/j.engstruct.2017.08.059.
65. Esfandiari MJ, Urgessa GS. Progressive collapse design of reinforced concrete frames using structural optimization and machine learning. Structures. 2020;28:1252–64. doi:10.1016/j.istruc.2020.09.039.
66. Duysak Y, Bekdaş G, Nigdeli SM. 2 Boyutlu betonarme çerçeve sistemin çiçek tozlaşma algoritmasi iile optimum tasarımı. In: 23. Ulusal Mekanik Kongresi; 2023 Sep 4–8; Konya, Turkıye. doi:10.1016/j.jobe.2022.104620.
67. Saedi Daryan A, Salari M, Palizi S, Farhoudi N. Size and layout optimum design of frames with steel plate shear walls by metaheuristic optimization algorithms. Structures. 2023;48:657–68. doi:10.1016/j.istruc.2022.11.118.
68. Lou H, Xiao Z, Wan Y, Quan G, Jin F, Gao B, et al. Size optimization design of members for shear wall high-rise buildings. J Build Eng. 2022;61:105292. doi:10.1016/j.jobe.2022.105292.
69. Esfandiari MJ, Urgessa GS, Sheikholarefin S, Dehghan Manshadi SH. Optimum design of 3D reinforced concrete frames using DMPSO algorithm. Adv Eng Softw. 2018;115:149–60. doi:10.1016/j.advengsoft.2017.09.007.
70. Mergos PE. Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. J Build Eng. 2021;44:102935. doi:10.1016/j.jobe.2021.102935.
71. ACI Committee 318. Building code requirements for structural concrete (ACI 318-19) and commentary (ACI 318R-19). Farmington Hills, MI, USA: American Concrete Institute; 2019.