



Article

Parameter-Free Metaheuristic-Based Method for Reinforced Concrete Frame Cost Optimization

Elmas Rakıcı Güldal ¹, Sinan Melih Nigdeli ¹, Gebrail Bekdas ^{1,*}  and Zong Woo Geem ^{2,*} 

¹ Department of Civil Engineering, Istanbul University-Cerrahpasa, 34320 Istanbul, Türkiye; elmas.rakiciguldal@ogr.iuc.edu.tr (E.R.G.); melihnig@iuc.edu.tr (S.M.N.)

² College of IT Convergence, Gachon University, Seongnam 13120, Republic of Korea

* Correspondence: bekdas@iuc.edu.tr (G.B.); geem@gachon.ac.kr (Z.W.G.)

Abstract

This study proposes a parameter-free metaheuristic optimization framework using the Jaya and Rao algorithms for the cost-based design of reinforced concrete (RC) frames in accordance with ACI 318-25. Beam and column dimensions are treated as design variables within predefined bounds, and the objective was to minimize the total construction cost including concrete and reinforcing steel. Structural analysis was performed using the matrix displacement method. The performance of the Jaya, Rao-1, Rao-2, and Rao-3 algorithms was evaluated through multiple independent runs. All methods achieved optimal or near-optimal solutions; however, Rao-2 demonstrated competitive performance with low mean values and favorable statistical performance. The results confirm the effectiveness of parameter-free metaheuristic methods for RC structural cost optimization. Unlike previous studies, this study explicitly focuses on parameter-free metaheuristic algorithms and evaluates their robustness through statistical analysis on reinforced concrete frame systems. The main contribution lies in demonstrating the comparative performance and practical applicability of parameter-free algorithms without the need for algorithm-specific parameter tuning.

Keywords: reinforced concrete frame; cost optimization; Jaya algorithm; Rao algorithms; ACI 318-25

1. Introduction

Structural cost optimization has gained increasing importance in civil engineering due to rising material prices and the demand for sustainable construction practices. Reinforced concrete (RC) frame systems are among the most widely used structural systems in building construction, and their economic design directly influences the overall project feasibility.

Optimization should provide the best practically applicable solution by utilizing design variables under a specified objective function. Numerous heuristic and metaheuristic algorithms have been developed for optimal structural design. Examples include teaching-learning-based optimization (TLBO) [1] and the flower pollination algorithm (FPA) [2].

One of the most important objectives of optimization is to minimize costs while ensuring structural safety, as human demands and needs generally favor designs that achieve the most advantageous and cost-effective solutions. Another key objective is to achieve the minimum weight and maximum strength while maintaining structural safety.

The cost optimization of reinforced concrete (RC) frame elements is also a widely studied topic. Akin and Saka [3] used the harmony search algorithm to determine the total



Academic Editor: Peng Zhang

Received: 25 March 2026

Revised: 23 April 2026

Accepted: 24 April 2026

Published: 26 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

cost of a frame, including concrete, formwork, and reinforcement steel costs for individual members, in accordance with the provisions of ACI 318-05.

Wang et al. [4] used a large-scale optimization algorithm called the reinforcement learning level-based particle swarm optimization algorithm (RLLPSO) in their study. They compared experimental results obtained from two large-scale benchmark test suites with five state-of-the-art large-scale optimization algorithms and showed that RLLPSO was superior in most cases. Regupathi [5] developed a genetic algorithm program in MATLAB for the cost optimization of RC beams and columns and applied it to multi-story RC frame structures. Bekdaş, Nigdeli, and Yang [6] conducted a study on the optimal design of RC members using the harmony search algorithm. Chutani and Singh [7] presented examples of RC frame optimization by hybridizing improved versions of particle swarm optimization (PSO) and the standard gravitational search algorithm (GSA). Ulusoy et al. [8] used various metaheuristic algorithms, such as compatibility search, bat algorithm, and teaching-learning-based optimization, in the design of various RC beams for cost minimization. Maaroo et al. [9] presented a hybrid framework integrating response surface methodology (RSM) with genetic algorithm (GA) to improve predictive modeling and optimize cement mixes. Their findings highlighted the robustness of the hybrid GA + RSM approach in capturing nonlinearities, improving accuracy, and achieving superior optimization results. Yang et al. [10] used three metaheuristic optimization algorithms, namely the antlion optimization algorithm (ALO), the moth flame optimization algorithm (MFO), and the salp swarm algorithm (SSA), to select the optimal hyperparameters of the random forest (RF) model to estimate the punching shear strength (PSS) of fiber-reinforced polymer-reinforced concrete (FRP-RC) beams. Hong et al. [11] proposed using artificial neural networks to approximate well-defined objective functions and other output parameters by a universal function, which can provide a generalized solution and enable the effective utilization of Jacobian and Hessian matrices in solving the Lagrangian function. Lin et al. [12] investigated recycled rubber aggregate concrete (RRAC) and applied a comprehensive framework integrating mechanical property prediction, sustainability assessment, and mix-design optimization. They trained models for compressive strength and elastic modulus using machine learning algorithms, identified the best-performing model, and employed particle swarm optimization to generate mixture designs that balance strength, stiffness, and carbon footprint, achieving significant emission reductions without compromising performance. Lin et al. [13] investigated the post-fire behavior of recycled aggregate concrete (RAC) containing coarse and fine aggregates. They tested 225 specimens at temperatures up to 800 °C, analyzed key mechanical properties, and proposed constitutive models and conversion relationships for residual strength and elastic modulus, providing insights into RAC performance after high-temperature exposure. Yu et al. [14] investigated size effects on uniaxial tensile and compressive behaviors of recycled aggregate concrete (RAC) produced using conventional and equivalent mortar volume (EMV) methods. They conducted experimental and numerical analyses, developing 2D random polygonal aggregate and discrete element mesomodels to evaluate the influence of the water-to-cement ratio, recycled aggregate content, and fabrication methods on compressive and tensile strengths, elastic modulus, and peak strains. The study proposed size reduction coefficients for practical application, showing that the EMV method improved compressive strength and elastic modulus, delayed compressive damage, and provided guidance for adjusting design parameters in RAC. Thomas et al. [15] investigated the strength and durability of concrete containing recycled concrete aggregates (RCA), showing that up to 25% of natural aggregates can be replaced without significantly affecting concrete performance. They also developed predictive mathematical models and a mixed design methodology for RCA concrete. Çakıroğlu et al. [16] investigated the use of the

Jaya metaheuristic algorithm to optimize reinforced concrete structures for both carbon emissions and cost while maintaining satisfactory structural performance. They applied the algorithm to determine optimal reinforcement ratios, concrete compressive strength, and cross-sectional dimensions at different levels of ultimate shear strength. The study demonstrated that optimization could significantly reduce both cost and carbon emissions without compromising structural safety. Gheyratmand et al. [17] presented an artificial bee colony algorithm (ABCA) for the optimum design of RC frames under combined gravity and lateral static loads. Bekdaş and Nigdeli [18] modified the harmony search algorithm for RC frames subjected to both static and dynamic loads.

Among the parameter-free optimization techniques, the Jaya algorithm and Rao algorithms have attracted considerable attention. These methods eliminate the need for algorithm-specific control parameters, simplifying implementation and improving robustness. The Jaya algorithm operates by iteratively moving candidate solutions toward the best solution while avoiding the worst solution. Similarly, the Rao-1, Rao-2, and Rao-3 algorithms utilize mathematical relationships between the best and worst individuals in the population to update design variables without requiring additional tuning parameters.

Although numerous studies have addressed the cost optimization of reinforced concrete structures using metaheuristic algorithms, most of them rely on parameter-dependent techniques that require the careful tuning of algorithm-specific parameters. This increases the implementation complexity and may affect the reliability of the results. In contrast, studies focusing specifically on parameter-free algorithms for reinforced concrete frame cost optimization are relatively limited and lack a comprehensive statistical evaluation of robustness. Therefore, this study aims to fill this gap by systematically evaluating parameter-free algorithms (Jaya and Rao variants) in terms of cost efficiency, convergence behavior, and robustness. Therefore, this study aims to evaluate and compare the performance of the Jaya and Rao algorithms for the cost optimization of a single-story two-span and three-story, three-span RC frame designed in accordance with ACI 318-25 [19] provisions.

The structural analysis was performed using the matrix displacement method coded in MATLAB 2018, and the results were verified using SAP2000. Statistical evaluation based on 20 independent runs was conducted to assess the robustness, convergence stability, and reliability of the algorithms.

2. Materials and Methods

The design and optimization procedures in this study were formulated in accordance with the provisions of ACI 318-25 [19]. The governing design expressions used in the developed computational framework are consistent with the current code provisions for the considered structural members. The detailed formulation of the structural analysis and optimization procedure are presented in the following subsections.

2.1. Structural Analysis: The Matrix Displacement Method

Structural analysis of frame systems requires the determination of internal forces and nodal displacements. Among the available approaches—force method, slope-deflection method, moment distribution method—the matrix displacement method (MDM) has been adopted in this study due to its efficiency for planar frames.

For a 2D frame element, each node has three degrees of freedom (DOFs): horizontal displacement Δx , vertical displacement Δy , and rotation θz . Thus, a two-node element has 6 DOFs in total. The global behavior of the structure is expressed in the global coordinate system, while the internal forces and moments of individual elements are determined in

their local coordinate system. Transformation between these systems is performed via the rotation matrix:

$$\begin{bmatrix} X_L \\ Y_L \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_G \\ Y_G \end{bmatrix} \quad (1)$$

In Equation (1), X_L , Y_L represent the local coordinate components, X_G , Y_G represent the global coordinate components, and θ represents the angle between the local and global coordinate systems.

The 6×6 transformation matrix $[T]$ for a frame element becomes:

$$T = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The local element stiffness matrix $[K_L]$ links internal forces to local displacements:

$$[f] = [K_L][\Delta_L] \quad (3)$$

where f represents the internal forces, and Δ_L represents the local displacements.

Global stiffness matrices $[K_G]$ for each element are computed as:

$$[K_G] = [T]^T \cdot [K_L] \cdot [T] \quad (4)$$

After assembling all elements, the system stiffness matrix $[K_{GS}]$ is obtained, and the global force vector $[F]$ and global node displacements $[\Delta_G]$ are solved:

$$[\Delta_G] = [K_{GS}]^{-1} \cdot [F] \quad (5)$$

Local displacements are then recovered:

$$[\Delta_L] = [T] \cdot [\Delta_G], [f] = [K_L][\Delta_L] \quad (6)$$

This procedure yields internal forces and moments for beams and columns used as inputs for design and optimization.

2.2. Optimization Method

2.2.1. Jaya Algorithm

In this study, the optimization problem was solved using the Jaya algorithm. The Jaya algorithm is a population-based metaheuristic optimization method proposed by Rao R. Venkata [20]. The main principle of the algorithm is to move candidate solutions toward the best solution in the population while simultaneously moving them away from the worst solution. The term “Jaya” originates from Sanskrit and means “victory”, reflecting the algorithm’s objective of continuously improving solutions during the optimization process.

One of the key advantages of the Jaya algorithm is that it does not require algorithm-specific control parameters. Therefore, only the population size and the maximum number of iterations need to be defined during implementation. This characteristic simplifies the application of the algorithm and facilitates its adaptation to various optimization problems.

During each iteration, all candidate solutions in the population are updated based on the best and worst solutions. The updating mechanism of the Jaya algorithm can be expressed as follows:

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|) \quad (7)$$

where:

At the i th iteration, let v_n denote the number of variables ($j = 1, 2, \dots, v_n$), p_n denote the population size ($k = 1, 2, \dots, p_n$) and $X_{(j,k,i)}$ represent the value of the j th variable for the k th population member. $r_{(1,j,i)}$ and $r_{(2,j,i)}$ are two random numbers generated for the j th variable during the i th iteration. The term $r_{(1,j,i)}(X_{(j,best,i)} - |X_{(j,k,i)}|)$ represents the tendency of a solution to move toward the best solution, whereas the term, $r_{(2,j,i)}(X_{(j,worst,i)} - |X_{(j,k,i)}|)$ represents the tendency of a solution to move away from the worst solution. In each iteration, the newly generated solution is evaluated using the objective function. If the new solution provides a better fitness value than the current solution, it replaces the existing one. This process continues until the predefined stopping criterion, such as the maximum number of iterations, is reached.

Due to its parameter-less structure, ease of implementation, and fast convergence characteristics, the Jaya algorithm has been widely applied in various engineering optimization problems.

2.2.2. Rao Algorithms

The Rao algorithms are population-based and parameter-free heuristic optimization algorithms proposed by Rao R. Venkata [21]. These algorithms do not require complex control parameters. Although the workflow of the three algorithms is similar, the movement equation used differs for each algorithm. The Rao algorithm equations were adopted from the study of Ravipudi and Keesari [22].

During the i th iteration, let $s_{n,m,i}$ denote the value of the n th variable of the m th solution. Let $s_{n,b,i}$ represent the value of the n th variable of the best solution, $s_{n,w,i}$ denote the value of the n th variable of the worst solution, and $s'_{n,m,i}$ represent the newly obtained value of $s_{n,m,i}$. The parameters $r_{1,n,i}$ and $r_{2,n,i}$ are random numbers within the interval $[0, 1]$ generated for the n th variable during the i th iteration.

Rao-1 Algorithm: In the Rao algorithms, each candidate solution in the Rao-1 algorithm is updated by considering the best and worst solutions within the population. The updating mechanism is based on the principle of moving the current solution toward the best solution while simultaneously moving it away from the worst solution.

The solution update equation of the Rao-1 algorithm can be expressed as follows:

$$s'_{n,m,i} = s_{n,m,i} + r_{1,n,i} (s_{n,b,i} - s_{n,w,i}) \quad (8)$$

This approach creates a guiding effect during the search process and enables rapid convergence in the solution space. However, since the exploration capability is limited, there is a possibility that the algorithm may become trapped in local optima.

Rao-2 Algorithm: The Rao-2 algorithm extends the Rao-1 algorithm by additionally incorporating the difference between two randomly selected solutions from the population into the solution updating process. This mechanism aims to enhance the search capability of the algorithm by increasing diversity in the solution space.

The solution update equation of the Rao-2 algorithm can be expressed as follows:

$$s'_{n,m,i} = s_{n,m,i} + r_{1,n,i} (s_{n,b,i} - s_{n,w,i}) + r_{2,n,i} (|s_{n,m,i} \text{ or } s_{n,l,i}| - |s_{n,l,i} \text{ or } s_{n,m,i}|) \quad (9)$$

In Equation (9), the third term on the right-hand side represents the interaction between the current solution (the m th solution) and a randomly selected solution (the l th solution) from the population. These two terms depend on the Z values of the current and randomly selected solutions.

If the Z value of the current solution is better than that of the randomly selected solution, the third term in Equation (9) is expressed as: $r_{2,n,i}(|s_{n,m,i}| - |s_{n,l,i}|)$. Similarly, if the Z value of the randomly selected solution is better than that of the current solution, the third term in Equation (9) is expressed as: $r_{2,n,i}(|s_{n,l,i}| - |s_{n,m,i}|)$. Thus, the Rao-2 algorithm exhibits a more balanced search behavior compared to Rao-1, particularly for complex optimization problems.

Rao-3 Algorithm: The Rao-3 algorithm simultaneously utilizes both the best–worst solution relationship and the differences between randomly selected solutions. This structure provides a more balanced optimization behavior between exploration and exploitation processes during the search.

The solution update equation of the Rao-3 algorithm can be expressed as follows:

$$s'_{n,m,i} = s_{n,m,i} + r_{1,n,i} (s_{n,b,i} - |s_{n,w,i}|) + r_{2,n,i} (|s_{n,m,i} \text{ or } s_{n,l,i}| - |s_{n,l,i} \text{ or } s_{n,m,i}|) \quad (10)$$

In Equation (10), the third term on the right-hand side represents the interaction between the current solution (the m th solution) and a randomly selected solution (the l th solution) from the population. These two terms depend on the Z values of the current and randomly selected solutions.

If the Z value of the current solution is better than that of the randomly selected solution, the third term in Equation (10) is expressed as: $r_{2,n,i}(|s_{n,m,i}| - |s_{n,l,i}|)$. Similarly, if the Z value of the randomly selected solution is better than that of the current solution, the third term is expressed as: $r_{2,n,i}(|s_{n,l,i}| - |s_{n,m,i}|)$. By incorporating both guiding and interaction mechanisms, the Rao-3 algorithm achieves a more balanced search performance compared to the previous variants.

3. Optimum Cost Design of Reinforced Concrete Frame System

In this section, the minimum-cost design of a single-story, two-span reinforced concrete frame is performed using both the Jaya and Rao optimization algorithms. To further validate the performance of the optimization algorithms, a three-story, three-span reinforced concrete frame system was analyzed. The design procedure was conducted in accordance with the provisions of ACI 318-25 (Building Code Requirements for Structural Concrete and Commentary) [19] published by the American Concrete Institute. A computational code was developed in the MATLAB environment to evaluate internal forces and perform the structural design. With appropriate modifications, the code can be extended to analyze systems with varying numbers of spans and stories, as well as to incorporate different optimization algorithms.

At the initial stage of the design process, key structural parameters such as the number of spans, number of stories, number of nodes, boundary conditions, element coordinates, and nodal connectivity are defined. The cross-sectional dimensions, namely the width (b) and height (h), are considered as design variables. These variables are randomly generated in the initial solution matrix within predefined lower and upper bounds to ensure compliance with design constraints.

Distributed live (L) and dead (D) loads are applied to the beams. The self-weight of the structure is excluded from the dead load and is instead calculated based on the randomly generated cross-sectional dimensions. This self-weight is then combined with the applied distributed loads.

The design constants considered in the analysis include the modulus of elasticity of steel (E_s), unit weight of steel (γ_s), unit weight of concrete (γ_c), clear concrete cover thickness (cc), compressive strength of concrete (f'_c), yield strength of steel (f_y), cost of concrete per cubic meter (C_c), and cost of steel per ton (C_s). The numerical values of these parameters are presented in Table 1.

Table 1. The design constants.

Description	Symbol	Unit	Value
Elasticity of steel	E_s	MPa	200,000
Unit weight of steel	γ_s	t/m ³	7.86
Unit weight of concrete	γ_c	t/m ³	2.5
Clear concrete cover thickness	cc	mm	40
Compressive strength of concrete	f'_c	MPa	30
Yield strength of steel	f_y	MPa	420
Cost of concrete per cubic meter	C_c	\$/m ³	130
Cost of steel per ton	C_s	\$/ton	1080

The single-story, two-span frame system is shown in Figure 1, while the three-story, three-span frame system is presented in Figure 2. The span lengths were taken as 6 m, and the story height as 3 m. The stopping criterion for the optimization (maximum number of iterations) was set to 100. The population size was taken as 20.

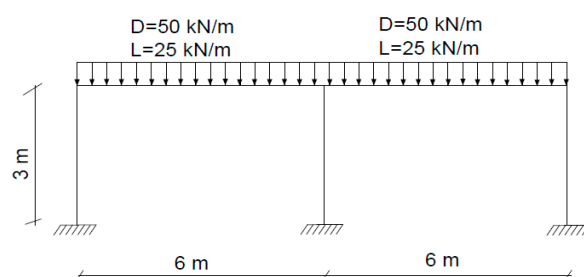


Figure 1. The single-story, two-span frame system.

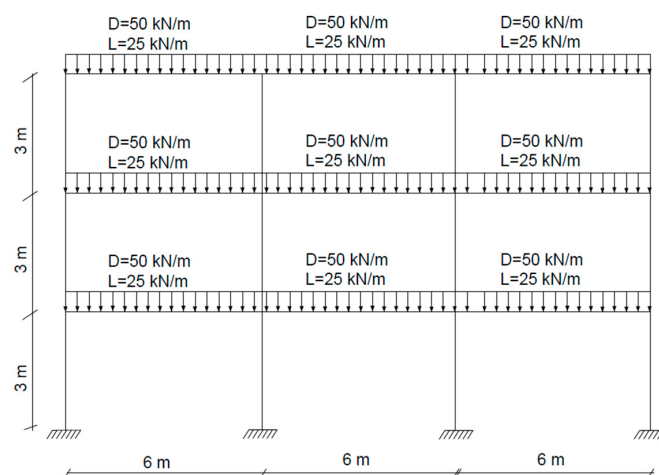


Figure 2. The three-story, three-span frame system.

The equations corresponding to the ACI-318 provisions are given below. In the design procedure, the reinforced concrete beams were designed first. The cross-sectional width (b) and height (h) were treated as design variables, constrained between 300 mm and 600 mm. It was assumed that under applied loads, compression and tension zones develop in the cross-section, separated by the neutral axis. In the compression zone, the actual stress

distribution in the concrete is replaced with an equivalent rectangular stress block that produces the same resultant force and location of action. The resultant compressive force in the concrete (F_c) and the resultant tensile force in the reinforcement (F_s) are given as:

$$F_c = 0.85f'_c ba \quad (11)$$

$$a = \beta_1 c \quad (12)$$

$$\begin{aligned} \beta_1 &= 0.85 \quad 17\text{MPa} < f'_c \leq 28\text{MPa} \\ \beta_1 &= 0.85 - 0.0071428(f'_c - 28) \quad f'_c > 28\text{MPa} \end{aligned} \quad (13)$$

$$F_s = A_s f_y \quad (14)$$

where c is the neutral axis depth, A_s is the reinforcement area, and the factor related to the depth of the equivalent rectangular stress block (β_1) is defined as β_1 in ACI 318 and shall not exceed 0.65. Beams were designed with under-reinforced conditions. The balanced reinforcement ratio (ρ_b) and the maximum reinforcement ratio (ρ_{max}) were calculated to ensure ductile behavior:

$$\rho_b = (0.85)\beta_1 \frac{f'_c}{f_y} \left(\frac{600}{600 + f_y} \right) \quad (15)$$

$$\rho_{max} = 0.75\rho_b \quad (16)$$

The modulus of elasticity of concrete (E_c) was calculated as:

$$E_c = 4700\sqrt{f'_c} \quad (17)$$

The analysis was performed using the load combination:

$$P = 1.4D + 1.6L \quad (18)$$

Minimum longitudinal reinforcement areas were ensured:

$$A_{s,min} \geq \frac{\sqrt{f'_c}}{4f_y} bd \quad (19)$$

$$A_{s,min} \geq \frac{1.4}{f_y} bd \quad (20)$$

Shear design of beams was conducted according to:

$$V_c = \frac{\sqrt{f'_c}}{6} bd \quad (21)$$

$$V_s = \frac{A_v f_y d}{s} \quad (22)$$

$$V_s = 0.66\sqrt{f'_c} bd \quad (23)$$

$$(A_v)_{min} = \frac{1}{3} \frac{bs}{f_y} \quad (24)$$

$$s \leq \begin{cases} d/4 \\ 8\phi_{min} \\ 150 \text{ mm} \end{cases} \quad (25)$$

In Equations (21)–(25), d is the effective depth of the beam obtained by subtracting the clear concrete cover from the overall beam height. V_c represents the nominal shear

strength of the concrete, V_s denotes the nominal shear strength of the reinforcement, and s is the spacing between transverse reinforcement. After all conditions are satisfied, the final values of A_v/s are calculated.

The material cost of each beam element was calculated as:

$$M_m = A_c * l_e * C_c + T_s * C_s \quad (26)$$

In Equation (26), l_e is the length of the element, T_s is the amount of reinforcement in tons, and A_c is the minimum gross cross-sectional area. After calculating the material cost for each beam element, the calculation proceeds to the material cost of the other structural element, namely the columns.

For columns, cross-sectional dimensions (b and h) were randomly generated as in beams and verified for minimum requirements. Column design used the moment amplification procedure from ACI 318. Top and bottom end factors, Ψ_A and Ψ_B , and the effective length factor for buckling, k , were computed as:

$$\Psi_{A,B} = \frac{\sum (EI/l)_{column}}{\sum (EI/l)_{beam}} \quad (27)$$

$$k = \frac{20 - \Psi_m}{20} \sqrt{1 + \Psi_m} \quad \text{if } \Psi_m < 2 \quad (28)$$

$$k = 0.9 \sqrt{1 + \Psi_m} \quad \text{if } \Psi_m \geq 2 \quad (29)$$

$$\Psi_m = 0.5(\Psi_A + \Psi_B) \quad (30)$$

Correction factor C_m and critical buckling load P_c were calculated as:

$$C_m = 0.6 + 0.4 \frac{M_1}{M_2} \quad (31)$$

$$P_c = \frac{\pi^2 EI}{(kl)^2} \quad (32)$$

$$\delta_s = \frac{C_m}{1 - \frac{P_u}{0.75 P_c}} \quad (33)$$

In Equation (31), M_1 and M_2 are the column end moments obtained from structural analysis, with M_2 taken as the larger value. The coefficient C_m must be greater than 0.4, and if any lateral load acts between the column ends, it should be taken as 1.0. In Equation (33), δ_s is the moment amplification factor, and P_u is the factored axial load. Minimum bending moment M_{min} was calculated as:

$$M_{min} = P_u(15 + 0.03h) \quad (34)$$

In Equation (34), both 15 and h are taken in mm. Thus, the design sectional forces are obtained.

For columns, the minimum reinforcement ratio $\rho_{min} = 0.01$ and the maximum reinforcement ratio $\rho_{maks} = 0.06$ are used, and the generated cross-sectional dimensions are multiplied to check the reinforcement area requirements. If the calculated sectional forces fall outside the usable region of the interaction diagram, the reinforcement area is penalized.

After calculating the longitudinal reinforcement area, the shear reinforcement design of the columns is performed. The shear reinforcement design is carried out in the same manner as for the beams.

The objective function representing the total material cost of the structure, OF, is defined as:

$$OF = \sum_{i=1}^n (M_m)_i \quad (35)$$

In Equation (35), n denotes the total number of elements.

For simplicity and consistency with the developed computational framework, the cost function includes only concrete and reinforcement costs. Although formwork cost is known to contribute significantly to total construction cost, it is excluded from this study. This simplification is not expected to significantly affect the comparative evaluation of the optimization algorithms, which is the primary objective of this research. Future studies may incorporate additional cost components for more comprehensive economic evaluation.

A flowchart, as shown in Figure 3, was developed to illustrate the optimization procedure used in this study. The diagram outlines the sequential steps, including calculating sectional forces for beams and columns, evaluating material costs, selecting the optimization algorithm (Jaya or Rao variants), and generating new candidate solutions. Decision points, such as checking whether a new solution improves the current best cost, are clearly indicated with Yes/No branches. This visual representation enhances clarity by providing a concise summary of the iterative process used for structural cost optimization.

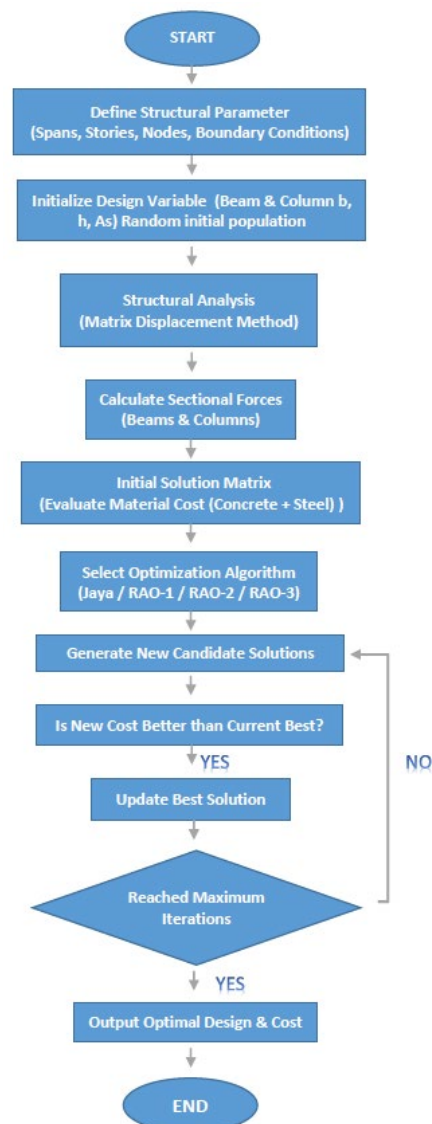


Figure 3. A flow diagram illustrating the step-by-step procedure for structural cost optimization.

4. Results and Discussion

The present study compared the performance of four parameter-free metaheuristic algorithms (Jaya, Rao-1, Rao-2, and Rao-3) in the cost optimization of a reinforced concrete frame designed as per ACI 318-25. The analysis was conducted through 20 independent runs for each algorithm to assess convergence stability and robustness. The cost results of 20 independent studies for a single-story, two-span reinforced concrete frame system are presented in Table 2.

Table 2. Results of cost optimization for a single-story two-span reinforced concrete frame.

JAYA	Rao-1	Rao-2	Rao-3
1170.3	1217.8	1170.3	1214.8
1214.8	1180	1170.3	1170.3
1217.8	1214.8	1170.3	1170.3
1170.3	1170.3	1170.3	1217.8
1170.3	1386.3	1170.3	1214.8
1217.8	1214.8	1217.8	1214.8
1170.3	1170.3	1170.3	1214.8
1170.3	1214.8	1170.3	1260.5
1214.8	1214.8	1217.8	1223.9
1217.8	1214.8	1214.8	1217.8
1214.8	1170.3	1217.8	1214.8
1170.3	1214.8	1217.8	1170.3
1635	1214.8	1170.3	1217.8
1170.3	1214.8	1170.3	1170.3
1170.3	1217.8	1214.8	1214.8
1170.3	1170.3	1214.8	1217.8
1170.3	1170.3	1217.8	1217.8
1214.8	1214.8	1214.8	1170.3
1214.8	1217.8	1170.3	1217.8
1170.3	1170.3	1170.3	1170.3

Based on these results, the minimum, maximum, mean values, and standard deviations were calculated and are shown in Table 3.

Table 3. Results of the comparison for a single-story two-bay reinforced concrete frame.

Algorithm	Minimum Cost (\$)	Maximum Cost (\$)	Mean Cost (\$)	Standard Deviation
Jaya	1170.3	1635.0	1182.3	37.2
Rao-1	1170.3	1386.3	1202.1	49.0
Rao-2	1170.3	1217.8	1181.0	15.7
Rao-3	1170.3	1260.5	1215.1	21.8

All algorithms successfully achieved the minimum cost value of 1170.3 in at least one run, indicating their effectiveness in locating the global optimum within the design space. However, the variability in the results differed among the algorithms.

The Jaya algorithm demonstrated a mean cost of 1182.3 with a relatively high standard deviation of 37.2, indicating a wider spread of solutions and occasional convergence to suboptimal points, as evidenced by a maximum cost of 1635. Notably, this high outlier increased the average cost and variability.

Rao-1 showed the highest average cost (1202.1) and standard deviation (49.0), along with a maximum cost of 1386.3, reflecting less consistent performance and a tendency to converge more slowly or to local minima compared to the others.

Rao-3 achieved a mean cost of 1215.1 and a moderate standard deviation of 21.8, displaying better stability than Jaya and Rao-1 but was still less consistent than Rao-2.

Among all, Rao-2 exhibited the lowest mean cost (1181.0) and the smallest standard deviation (15.7), alongside the narrowest cost range (1170.3 to 1217.8). This indicates that Rao-2 provides relatively better robustness and convergence consistency for the single-story, two-span frame case.

In addition to the single-story, two-span reinforced concrete frame system, twenty independent analyses were performed for the three-story, three-span reinforced concrete frame system. The results are presented in Table 4.

Table 4. Results of the cost optimization for a three-story three-span reinforced concrete frame.

JAYA	Rao-1	Rao-2	Rao-3
3055.84	2838.27	2743.59	2983.66
2995.56	2827.88	2952.48	2886.87
2980.87	2842.38	2785.97	2828.69
2975.03	2863	2894.32	2865.41
2871.79	2862.51	2757.97	2923.17
2860.85	3005.08	2727.87	3053.89
2841.17	2908.39	2964.8	2893.67
2841.17	2834.41	2915.73	2834.41
2849.32	2959.46	2741.27	2827.88
2866.96	2846.07	2838.28	2824.99
2980.87	2842.38	2785.97	2828.69
2975.03	2863	2894.32	2865.41
2871.79	2862.51	2757.97	2923.17
2860.85	3005.08	2727.87	3053.89
3055.84	2908.39	2743.59	2983.66
2995.56	2838.27	2952.48	2886.87
2980.87	2827.88	2785.97	2828.69
2975.03	2842.38	2964.8	2865.41
2860.85	2908.39	2915.73	2827.88
2841.17	2834.41	2741.27	2824.99

Based on these results the best, worst, mean solutions, and standard deviation values are summarized in Table 5.

Table 5. Results of the comparison for a three-story three-span reinforced concrete frame.

Algorithm	Minimum Cost (\$)	Maximum Cost (\$)	Mean Cost (\$)	Standard Deviation
Jaya	2841.17	3055.84	2922.5	78.6
Rao-1	2827.88	3005.08	2875.6	55.4
Rao-2	2727.87	2964.80	2836.2	87.1
Rao-3	2824.99	3053.89	2872.9	79.8

The statistical results are summarized as follows: JAYA achieved a best cost of 2841.17 and a worst cost of 3055.84, with a mean value of approximately 2922.5 and a standard deviation of 78.6. Rao-1 produced a best solution of 2827.88 and a worst solution of 3005.08, yielding an average cost of about 2875.6 with a relatively low standard deviation of 55.4, indicating stable performance.

Rao-2 demonstrated the best overall minimum cost (2727.87) among all algorithms, although it exhibited higher variability with a standard deviation of 87.1 and a maximum cost of 2964.80. Rao-3 resulted in a minimum cost of 2824.99 and a maximum of 3053.89, with an average cost of 2872.9 and a standard deviation of 79.8.

Overall, Rao-2 appeared to be the most effective algorithm in terms of achieving the lowest cost, while Rao-1 provided more consistent results with lower variability. Jaya, although competitive, showed comparatively higher average cost values. Although Rao-2 achieved the lowest minimum cost among all algorithms, it exhibited the highest standard deviation, indicating significant variability in its performance for more complex structural systems. This suggests that while Rao-2 is effective in exploring the solution space, its consistency decreases as the problem complexity increases. In contrast, Rao-1 demonstrated more stable behavior with lower variability. Therefore, the robustness of the algorithms appears to depend on the complexity of the structural system.

5. Conclusions

The results indicate that the performance of parameter-free algorithms varies depending on the complexity of the structural system. While Rao-2 demonstrates strong performance and robustness for simpler problems, its variability increases for more complex frames. In such cases, Rao-1 provides more stable and consistent results. Therefore, no single algorithm can be considered universally superior, and the selection should depend on the characteristics of the structural optimization problem. These findings affirm the potential of parameter-free metaheuristics for reinforced concrete structural optimization, providing practitioners with efficient alternatives that do not require algorithm-specific parameter tuning.

The convergence behavior of the JAYA, Rao-1, Rao-2, and Rao-3 algorithms was analyzed for both the single-story, two-span and the three-story, three-span reinforced concrete frame systems. Figure 4 presents the convergence curves obtained from twenty independent runs for the best algorithm, illustrating the progression of the best cost value over successive iterations.

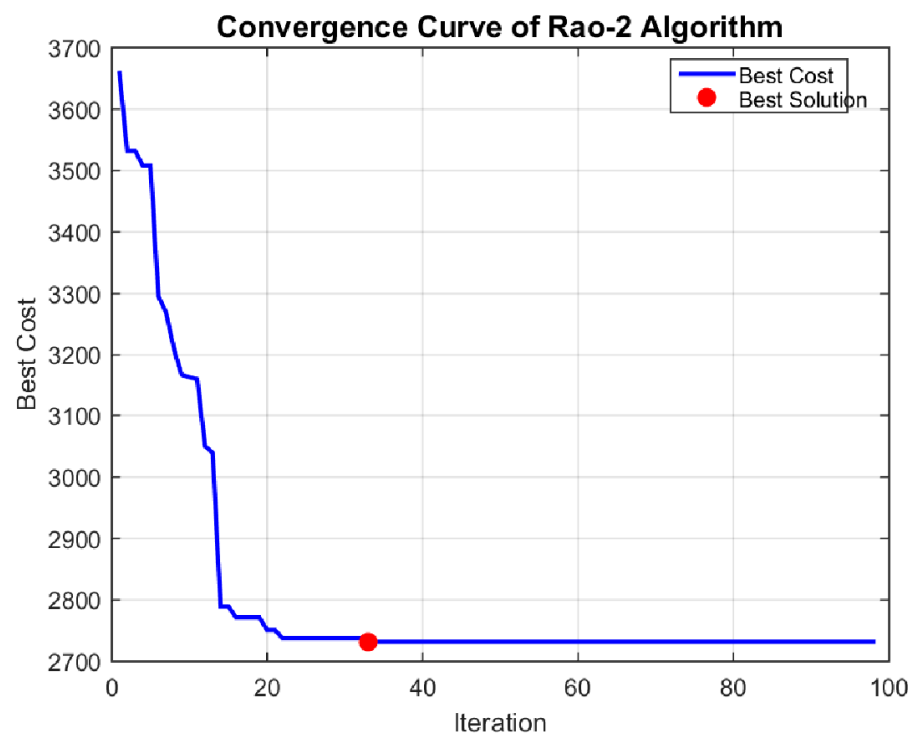


Figure 4. The convergence curve of the Rao-2 algorithm for a three-story, three-bay reinforced concrete frame system.

The convergence behavior of the Rao-2 algorithm was analyzed over 100 iterations with a population size of 20. The evolution of the best cost value across iterations is illustrated in the convergence curve.

As observed, the algorithm demonstrated a rapid decrease in the objective function value during the early stages of the search process. The initial cost started at approximately 3658 and dropped significantly within the first 15 iterations, reaching around 2789. This indicates an effective exploration capability of the algorithm in the initial phase.

Between iterations 15 and 30, the algorithm continued to improve the solution, albeit at a slower rate, eventually reaching a cost value of approximately 2737. After iteration 30, the convergence curve became relatively stable, with only minor improvements observed. The algorithm reached its best cost value of approximately 2727.87 around iteration 33.

From this point onward, the cost value remained constant until the final iteration, indicating that the algorithm had converged to a near-optimal solution. This stable behavior demonstrates the strong exploitation capability of the Rao-2 algorithm in the later stages of the optimization process.

Overall, the convergence curve reveals that the Rao-2 algorithm achieves a good balance between exploration and exploitation, with fast initial convergence followed by stable refinement of the solution.

Author Contributions: Conceptualization, S.M.N. and E.R.G.; methodology, E.R.G. and G.B.; software, E.R.G.; validation, S.M.N.; formal analysis, E.R.G.; investigation, E.R.G.; writing—original draft preparation, E.R.G.; writing—review and editing, S.M.N., Z.W.G. and G.B.; visualization, E.R.G.; supervision, G.B. and Z.W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Crepinsek, M.; Liu, S.; Mernik, L. A note on teaching–learning-based optimization algorithm. *Inf. Sci.* **2012**, *212*, 79–93. [[CrossRef](#)]
2. Balasubramani, K.; Marcus, K. A study on flower pollination algorithm and its applications. *Int. J. Appl. Innov. Eng. Manag.* **2014**, *3*, 230–235.
3. Akin, A.; Saka, M.P. Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318-05 provisions. *Comput. Struct.* **2015**, *147*, 79–95. [[CrossRef](#)]
4. Wang, F.; Wang, X.; Sun, S. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Inf. Sci.* **2022**, *602*, 298–312. [[CrossRef](#)]
5. Regupathi, R. Cost Optimization of Multistoried Rc Framed Structure Using Hybrid Genetic Algorithm. *Int. Res. J. Eng. Technol.* **2017**, *4*, 888–896.
6. Bekdaş, G.; Nigdeli, S.M.; Yang, X. Optimum reinforced concrete design by harmony search algorithm. In *Metaheuristics and Optimization in Civil Engineering*; Springer: Cham, Switzerland, 2016; pp. 165–180.
7. Chutani, S.; Singh, J. Use of modified hybrid PSOGSA for optimum design of RC frame. *J. Chin. Inst. Eng.* **2018**, *41*, 342–352. [[CrossRef](#)]
8. Ulusoy, S.; Kayabekir, A.E.; Bekdaş, G.; Niğdeli, S.M. Metaheuristic algorithms in optimum design of reinforced concrete beam by investigating strength of concrete. *Chall. J. Concr. Res. Lett.* **2020**, *11*, 26–30. [[CrossRef](#)]
9. Maaroo, R.S.; Hamad, A.S.; Hussein, M.M.F. A novel hybrid metaheuristic algorithm and response surface methodology approach for predictive modeling and optimization of cementitious compressive strength. *Sci. Rep.* **2025**, *16*, 2894. [[CrossRef](#)] [[PubMed](#)]
10. Yang, P.; Li, C.; Qiu, Y.; Huang, S.; Zhou, J. Metaheuristic optimization of random forest for predicting punch shear strength of FRP-reinforced concrete beams. *Materials* **2023**, *16*, 4034. [[CrossRef](#)] [[PubMed](#)]
11. Hong, W.K.; Nguyen, M.C. AI-based Lagrange optimization for designing reinforced concrete columns. *J. Asian Archit. Build. Eng.* **2022**, *21*, 2330–2344. [[CrossRef](#)]

12. Lin, L.; Xu, N.; Yang, D.; Li, G.; Xiao, Y.; Yu, Y. A smart computational framework for predicting mechanical and sustainability indicators and optimizing mix proportions of recycled rubber aggregate concrete. *Eng. Appl. Artif. Intell.* **2026**, *164*, 113364. [[CrossRef](#)]
13. Lin, L.; Xu, J.; Ying, W.; Yu, Y.; Zhou, L. Post-fire compressive mechanical behaviors of concrete incorporating coarse and fine recycled aggregates. *Constr. Build. Mater.* **2025**, *461*, 139948. [[CrossRef](#)]
14. Yu, Y.; Zhou, L.; Khodadadi, N.; Nanni, A. Size Effects on Uniaxial Compression and Tension in Recycled Aggregate Concrete Fabricated Using Conventional and Equivalent Mortar Volume Techniques. *J. Mater. Civ. Eng.* **2026**, *38*, 04026051. [[CrossRef](#)]
15. Thomas, J.; Thaickavil, N.N.; Wilson, P.M. Strength and durability of concrete containing recycled concrete aggregates. *J. Build. Eng.* **2018**, *19*, 349–365. [[CrossRef](#)]
16. Cakiroglu, C.; Islam, K.; Bekdas, G.; Apak, S. Cost and CO2 emission-based optimisation of reinforced concrete deep beams using Jaya algorithm. *J. Environ. Prot. Ecol.* **2022**, *23*, 2420–2429.
17. Gheytratmand, C.; Gholizadeh, S.; Vababzadeh, B. Optimization of RC frames by an improved artificial bee colony algorithm. *Int. J. Optim. Civ. Eng.* **2015**, *5*, 189–203.
18. Bekdaş, G.; Nigdeli, S.M. Modified Harmony Search for Optimization of Reinforced Concrete Frames. In *Harmony Search Algorithm: Proceedings of the 3rd International Conference on Harmony Search Algorithm (ICHSA 2017)*; Springer: Singapore, 2017; pp. 213–221.
19. ACI 318-25; Building Code for Structural Concrete—Code Requirements and Commentary and Commentary. American Concrete Institute: Farmington Hills, MI, USA, 2025.
20. Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [[CrossRef](#)]
21. Rao, R.V. Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *Int. J. Ind. Eng. Comput.* **2020**, *11*, 107–130. [[CrossRef](#)]
22. Ravipudi, V.R.; Keesari, H.S. *Design Optimization of Renewable Energy Systems Using Advanced Optimization Algorithms*; Springer: Cham, Switzerland, 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.