Jos Jornal of Applied Animal Science and Environmental Stability

Department of Animal Science. University of Jos.

Optimization of Reinforced Concrete Beam and Column Sections Design Using a Python-Based Genetic Algorithm Integrated with BIM Parametric Modelling: A Framework for Sustainable Land Development

Emmanuel Ushahemba Orhiiga[1], M.I. Alfa[1], J.E. Sani[1]

[1]*Department of Civil Engineering, Faculty of Engineering, University of Jos, Nigeria*

emmanuelorhiiga@gmail.com

Abstract

Conventional reinforced concrete (RC) structural design relies on manual procedures that produce over-designed, resource-intensive structures, driving up construction costs and embodied carbon per unit of developed land. This paper presents a Python-based Genetic Algorithm (GA) integrated with Building Information Modelling (BIM) parametric modelling as a unified, automated framework for the optimization of RC beam and column sections design in accordance to Eurocode. The framework employs real-valued GA encoding, penalty-based fitness evaluation, Simulated Binary Crossover (SBX), polynomial mutation, and bidirectional JSON-based data exchange between the optimizer (Python GA engine) and a parametric Autodesk Revit model driven by Dynamo scripting. A four-storey RC framed building serves as the case study, incorporating eleven mixed continuous-discrete design variables and twelve Eurocode-based structural constraints. The primary objective for the optimization is to minimize the cost of construction material subject to the code constraints with structural efficiency and estimated embodied CO_2 monitored as secondary performance metrics. The proposed framework

addresses a critical gap identified in the literature: the absence of a transparent, Python-based, bidirectionally integrated BIM-GA system with full design code compliance for RC member optimization. By systematically reducing material consumption per structural element, the framework directly contributes to cost-effective, sustainable land development, particularly relevant to rapidly urbanizing emerging economies where infrastructure construction costs constitute a primary barrier to viable land utilization. The framework is designed for practical deployment using widely accessible tools, enhancing adoption potential in small and medium engineering practices in developing contexts.

Keywords: Genetic Algorithm; Reinforced Concrete Optimization; Building Information Modelling; Eurocode 2; Structural Optimization; Land Development; Python; Revit; Dynamo

1. Introduction

Land development comprising residential, commercial, and infrastructure construction is among the most capital-intensive processes in modern economies. The economic viability of any land development project is intimately tied to construction costs, which in reinforced concrete (RC) dominated construction environments are substantially determined by the quantity and specification of structural materials consumed. The design of beams and columns, which constitute the primary load-carrying skeleton of RC-framed buildings, directly governs material demand, construction cost, and embodied environmental impact. In rapidly urbanizing emerging economies such as Nigeria, where infrastructure deficits are substantial and land development costs constitute a critical barrier to housing supply and economic expansion, even marginal reductions in structural material consumption translate into significant improvements in development feasibility and affordability (Benjamin, 2025).

Despite the availability of well-established design standards such as Eurocode 2 (CEN, 2004), conventional RC design practice remains constrained by its reliance on manual, iterative, and experience-dependent procedures. These methods are fundamentally unequipped to explore large multi-dimensional design spaces systematically, cannot simultaneously evaluate competing performance objectives, and typically result in conservative designs that consume more material than

structurally necessary (Cruz & Lemonge, 2020; Menezes et al., 2023). The construction industry contributes approximately 8% of global CO_2 emissions driven largely by cement production. This means that this systematic overdesign carries environmental consequences that are increasingly untenable (Alsabbagh & Abdulrahman, 2025).

Genetic Algorithms (GAs) in particular, as one of the Nature-inspired metaheuristic algorithms have emerged as powerful tools for navigating the complex, nonlinear, and multi-modal design landscapes characteristic of RC structural optimization. GAs employ population-based evolutionary search, consisting of selection, crossover, and mutation operators, to identify optimal or near-optimal solutions across large discrete and continuous design spaces (Goldberg, 1989; Holland, 1992). Parallel to these computational advances, Building Information Modelling (BIM) has transformed structural design practice by enabling the creation, management, and exchange of comprehensive parametric digital models integrating geometric, material, analytical, and project management information within one environment (Eastman et al., 2011; Azhar, 2011).

The convergence of Python-based computation, GA optimization, and BIM API connectivity has opened a powerful frontier in automated structural design. However, a critical integration gap persists: BIM environments lack inherent optimization intelligence; optimization algorithms typically operate in isolation from BIM platforms; and data exchange between these systems remains predominantly unidirectional and manual. This undermines automation, scalability, and practical usability (Machado et al., 2021; Afzal et al., 2023). This paper directly addresses this gap by presenting a transparent, bidirectional, fully automated framework integrating Python-based Genetic Algorithms with BIM parametric modelling for the systematic optimization of RC beam and column sections design under Eurocode 2, with explicit application to the context of sustainable land development.

The specific contributions of this paper are: the formulation of a mixed continuous-discrete optimization problem for RC beam and column sections with eleven design variables and twelve Eurocode 2-compliant constraints; the development and implementation of a GA using real-valued encoding, SBX crossover, polynomial mutation, and penalty-based constraint handling in Python; a

bidirectional BIM integration framework using Autodesk Revit and Dynamo with JSON-based data exchange; and a demonstration of how this framework supports cost-effective and sustainable land development outcomes through systematic structural material optimization.

2. Literature Review

2.1 Structural Optimization and Land Development

Structural optimization is the systematic process of identifying the most efficient design configuration while satisfying prescribed safety, serviceability, code compliance, and economic constraints. In the context of land development, its significance is amplified by the direct relationship between structural material efficiency and overall development cost. Ranjan and Singh (2025) demonstrated that GA-integrated BIM optimization of a G+3 RC building achieved 14.2% reductions in concrete volume and 12% reductions in structural cost relative to conventional manual design, reductions that translate directly into lower per-unit development costs. Sherif et al. (2022) validated a BIM-based optimization model across eleven case studies, achieving material cost reductions of up to 15% per floor. At the environmental level, Alsabbagh and Abdulrahman (2025) showed that optimized RC slab designs can reduce CO_2 emissions by up to 29% relative to conventional configurations.

Structural optimization techniques are broadly classified into sizing, shape, and topology optimization. Sizing optimization is determining optimal cross-sectional properties such as beam depth and width, or column dimensions and reinforcement areas. It is the most practically relevant form for RC design and constitutes the primary focus of this study (Coello et al., 1997; Menezes et al., 2023; Cruz & Lemonge, 2020). Classical mathematical programming methods, while theoretically rigorous, are restricted to convex, differentiable problem formulations and struggle with the discrete variable spaces inherent in practical RC design. Metaheuristic methods, including Genetic Algorithms, Particle Swarm Optimization, and Simulated Annealing, overcome these limitations through population-based, derivative-free search, making them far better suited to the mixed-variable, multi-constraint landscape of RC structural design (Yang, 2010).

2.2 Genetic Algorithms in RC Structural Optimization

Genetic Algorithms have demonstrated well-established effectiveness in RC structural optimization across multiple structural element types and loading conditions. Coello et al. (1997) reported one of the earliest GA applications to RC beam design, producing cost-optimal cross-sections while satisfying ACI code constraints. Cruz and Lemonge (2020) demonstrated GA-based optimization of RC column cross-sections under biaxial bending, confirming that commercial bar diameters and discrete member dimensions, essential for constructability can be directly enforced within the optimization, producing feasible solutions at lower material cost than deterministic methods. Menezes et al. (2023) further confirmed that GA populations converge progressively toward cost-optimal column geometries, with larger initial populations proving essential for identifying global rather than local cost minima, recommending population sizes of 240-280 individuals for RC column optimization.

Ahmad et al. (2024), in a comprehensive bibliometric review of machine learning and metaheuristic applications in RC frame design from 1997 to 2023, identified GAs as one of the most consistently effective approaches across a range of structural types and loading conditions, while observing a clear escalation in research interest, from approximately ten publications annually in 2010 to over ninety by 2022. Most recently, Cabrera et al. (2025) demonstrated that Python-scripted structural modelling of a 17-storey RC building through the ETABS API achieved $R^2 = 0.995$ accuracy relative to manual modelling while completing the full process in just 13.2 minutes, with subsequent GA optimization achieving a 4.01% reduction in reinforcement demand, directly validating the computational rationale of Python-based GA-BIM integration.

2.3 BIM Integration with Structural Optimization

BIM enables the creation, management, and exchange of comprehensive digital representations of facilities throughout their lifecycle, integrating geometric, material, analytical, and project management information within a unified parametric model (Eastman et al., 2011). Platforms such as Autodesk Revit, extended by visual programming environments like Dynamo, provide infrastructure

for parametric design workflows in which structural element geometries and properties are governed by algorithmic rules rather than fixed manual inputs, enabling rapid generation and evaluation of multiple design alternatives (Sacks et al., 2018; Woodbury, 2010). BIM also enables automated quantity extraction and cost assessment directly from the model, ensuring that material efficiency gains identified through optimization are accurately captured rather than analytically approximated.

However, existing BIM-optimization integration frameworks are predominantly unidirectional and manual. This requires engineers to manually transfer optimization results into BIM models, introducing inefficiency, potential transcription error, and a fundamental break in the automation loop (Machado et al., 2021). Afzal et al. (2023), in a PRISMA systematic review, proposed an early-stage sustainable structural design optimization (ESSDO) framework integrating Revit, Robot Structural Analysis, and Dynamo, but one that has not been validated for RC beam and column sizing under full Eurocode 2 constraint compliance. The present study directly addresses this validation gap, proposing a transparent, Python-based, bidirectionally integrated BIM-GA framework tailored to the practical needs of land development engineering in emerging economy contexts.

3. Methodology

3.1 Case Study and Problem Definition

The case study is a representative four-storey RC framed building, comprising sixteen columns and twenty-four primary beams per floor (totaling sixty-four columns and ninety-six beams). A regular 4.0 m × 3.0 m bay grid with a 3.5 m storey height is adopted, representing a typical residential or mixed-use building typology in Nigerian land development contexts. Structural demands are computed using Eurocode 2 moment coefficients and tributary area methods, with gravity loading per Eurocode 1 (CEN, 2002) comprising permanent actions (self-weight and finishes) and variable actions (imposed floor loads). The case study scope is deliberately constrained to gravity loading at the preliminary design stage, enabling clear demonstration of the framework's optimization capabilities within a transparent, reproducible problem definition.

Material properties are specified per Eurocode 2: concrete grades C20/25 to C40/50 (characteristic compressive strengths f_{ck} = 20-40 MPa), reinforcement yield strength f_{yk} = 460 MPa and a partial material factor for concrete γ_c = 1.5 and for steel γ_s = 1.15. Unit material costs are based on representative Nigerian construction market rates.

3.2 Optimization Problem Formulation

The optimization problem is formulated as a mixed-variable, constrained minimization. The objective function minimizes total construction material cost, encompassing concrete volume cost, steel reinforcement cost, and formwork cost:

$$f(x) = C_c \cdot V_c(x) + C_s \cdot W_s(x) + C_f \cdot A(x) \quad \ldots\ldots\ldots(1)$$

where C_c, C_s, and C_f are unit costs of concrete, steel, and formwork, respectively, and $V_c(x)$, $W_s(x)$, and $A_f(x)$ are the total concrete volume, steel mass, and formwork area determined by the design vector x.

The beam design vector comprises five variables: width $b_w \in [200, 400]$ mm, total depth $h \in [350, 700]$ mm, concrete grade index $if_{ck} \in \{0,1,2,3\}$, longitudinal bar count $n_s \in \{2,...,6\}$, and bar diameter index $i\varphi_{bar} \in \{0,1,2,3\}$ (corresponding to 16, 20, 25, 32 mm). The column design vector comprises six variables: section width b, depth h, concrete grade index if_{ck}, longitudinal bar count n_{sc}, bar diameter index $i\varphi_{sc}$, and link diameter index $i\varphi_{tie}$. Combined, the optimization encompasses eleven mixed continuous-discrete design variables per member class.

Twelve Eurocode 2-compliant structural constraints are enforced: (i) flexural capacity design moment $M_{Ed} \leq M_{Rd}$; (ii) shear capacity — $V_{Ed} \leq V_{Rd}$; (iii) deflection serviceability — span/effective depth ratio check; (iv) minimum and maximum longitudinal reinforcement ratios ($0.0013 \leq \rho \leq 0.04$); (v) axial-bending interaction for columns — design point within the P-M interaction diagram; (vi) slenderness check and second-order eccentricity per EC2 Clause 5.8.3; (vii) minimum link spacing; (viii) minimum concrete cover; (ix) section aspect ratio limits; (x) minimum beam width for bar

accommodation; (xi) reinforcement spacing compliance; and (xii) constructability: discrete bar diameters from the commercially available set.

3.3 Genetic Algorithm Implementation

The GA is implemented in Python 3.10, utilizing NumPy (v1.24) for vectorized array operations, SciPy (v1.11) for supplementary mathematical functions, and the DEAP library (v1.4) for foundational GA infrastructure. Custom modules handle structural analysis functions: beam_design.py (flexural and shear capacity, deflection), column_design.py (P-M interaction diagram, slenderness), cost_model.py (objective function), and bim_interface.py (Revit API data exchange). The modular architecture ensures scalability and future extensibility to surrogate-modelling or reinforcement learning approaches.

Real-valued encoding is adopted, with chromosomes representing beam and column designs as vectors of continuous and integer-indexed discrete values. A population of N = 200 individuals is initialized using uniform random sampling across the full feasible range of each design variable, following the recommendation of Menezes et al. (2023) that larger, diverse initial populations are essential for locating global rather than local cost minima.

The penalized fitness function integrates cost minimization and constraint enforcement:

$$F(x) = f(x) \cdot (1 + \alpha \cdot P(x)) \quad \text{.....(2)}$$

where $\alpha = 10$ is the penalty coefficient and $P(x) = \Sigma \max(0, g_i(x))^2$ is the normalized constraint violation measure, with $g_i(x) = (\text{demand} - \text{capacity})/\text{capacity}$ for each capacity-type constraint. This formulation ensures infeasible solutions incur elevated costs proportional to their degree of violation while retaining them in the population as potential sources of diversity.

Tournament selection (tournament size $k_t = 3$) is employed as the selection mechanism, preferred over roulette wheel selection for its resistance to premature convergence when fitness variance is low and its consistent selection pressure independent of absolute fitness scale (Mitchell, 1996). Simulated Binary Crossover (SBX, distribution index $\eta_c = 20$, probability $P_c = 0.9$) is applied to continuous

variables; uniform crossover is applied to discrete integer variables. Polynomial mutation (distribution index $\eta_m = 20$, probability $P_m = 1/n_{vars}$) introduces diversity in continuous variables; random resetting mutation is applied to discrete variables. Elitism retains the single best individual from each generation, guaranteeing monotonic improvement of best fitness across generations.

The GA terminates when: (i) a maximum of 150 generations is completed; (ii) improvement in best fitness over 30 consecutive generations falls below 0.1%; or (iii) population standard deviation falls below 0.5% of mean population fitness. The algorithm is executed 30 independent times to verify robustness and confirm global rather than local optima, with solution quality assessed by coefficient of variation across runs (CV < 5% target).

3.4 BIM-Based Parametric Modelling and Integration

The BIM component is developed in Autodesk Revit 2024 with Dynamo v2.17 as the scripting interface. Beam families expose six instance parameters; width, total depth, concrete grade, bar diameter, bar count, and link spacing, mapped directly to GA design variables. Column families expose analogous parameters. All analytical properties (member connectivity, boundary conditions, load assignments) are defined within the Revit analytical model to maintain consistency between the geometric BIM record and the GA's structural computations. Material assets are linked to Revit material objects, enabling automated propagation of compressive strength, elastic modulus, and density in response to GA-driven grade changes.

Bidirectional data exchange is implemented via JSON file transfer, a lightweight, platform-independent, and auditable mechanism. The GA writes candidate parameter sets to a JSON output file; Dynamo reads, applies these parameters to the Revit model via Element.SetParameterByName nodes, and returns updated material takeoff metrics (concrete volumes, steel masses, formwork areas) in a feedback JSON file. This protocol avoids in-process API coupling and its associated version-compatibility risks (Naser et al., 2022).

Each optimization generation proceeds through six automated steps: (1) GA generates and decodes candidate chromosomes; (2) decoded parameters are written to JSON exchange file; (3) Dynamo

reads and updates all structural element families in Revit via API; (4) GA evaluates structural performance using the Python design modules; (5) for elite solutions, Dynamo extracts BIM-based material takeoffs to validate analytical cost computations; and (6) selection, crossover, and mutation operators generate the next generation. Upon convergence, optimized parameters are permanently written to the Revit model, enabling direct generation of construction documentation, reinforcement schedules, and material takeoff reports from the definitive BIM record. Figure 1 presents the complete workflow architecture.

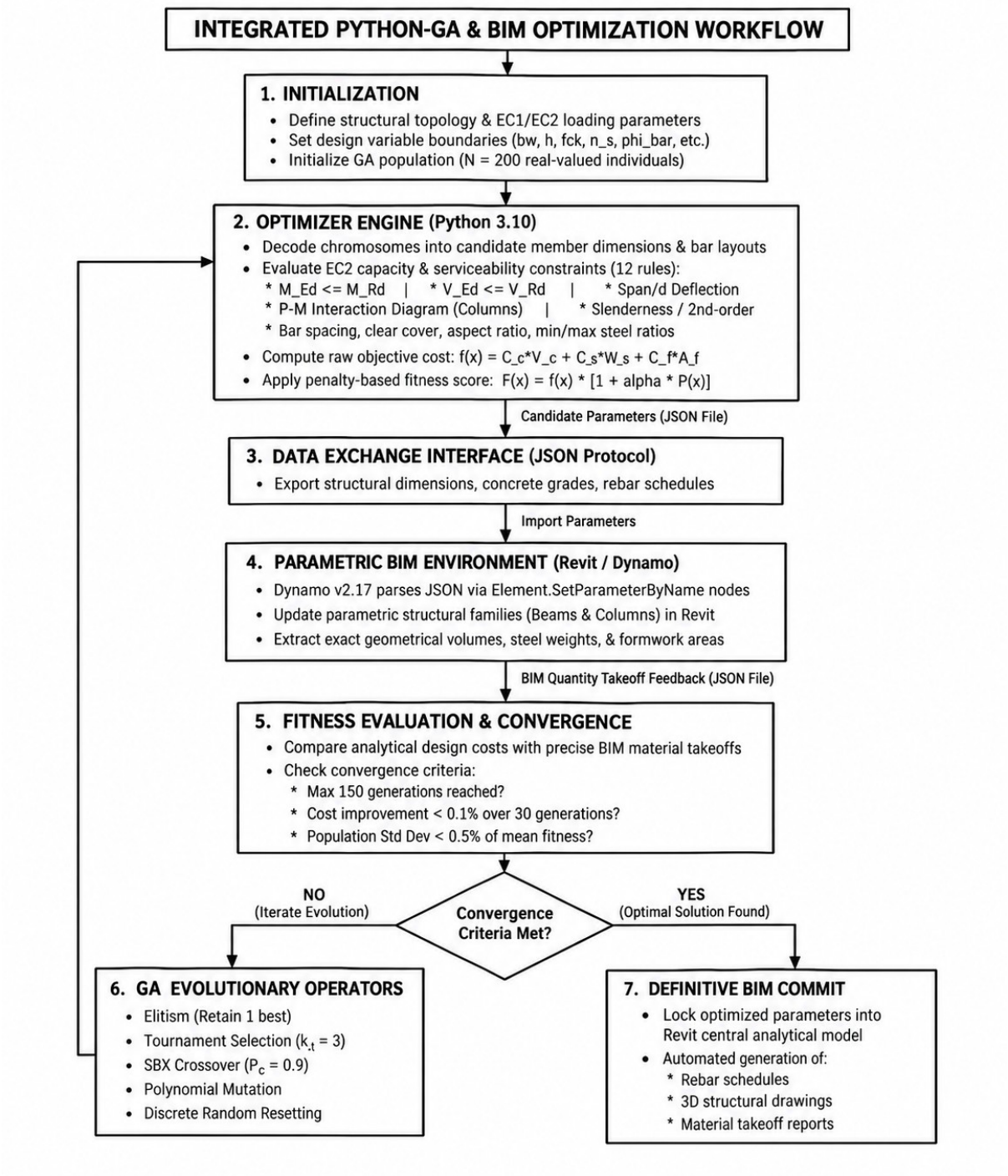


Figure 1. Integrated GA-BIM Framework Workflow for RC Member Optimization

4. Framework Analysis and Expected Outcomes

4.1 Implications for Land Development Cost

The direct relevance of structural optimization to land development economics merits explicit examination. In a typical four-storey RC-framed building, the case study typology employed in this

study, structural materials (concrete and reinforcement steel) account for approximately 35-45% of total construction cost in Nigerian practice. A conservative 10% reduction in structural material cost, consistent with the lower bound of documented GA optimization savings in comparable studies (Ranjan & Singh, 2025; Sherif et al., 2022), would therefore reduce total building construction cost by approximately 3.5-4.5%. For a development comprising residential units at a typical construction cost of ₦45 million per unit, this translates to potential savings of approximately ₦78.75-101.25 million, a substantial improvement in development viability.

Beyond direct cost savings, structural optimization has second-order land development benefits that are frequently underacknowledged. Reduced structural section sizes, achievable when sections are sized at their true structural optima rather than conservatively, increase net lettable floor area per structural bay, improving the efficiency of land utilization. Smaller column sections, for instance, directly increase net floor area without altering the building footprint, enhancing the economic return per unit of land developed. These combined effects, direct material cost reduction and improved space efficiency position structural optimization as a core enabler of financially viable and sustainable land development.

4.2 Novelty and Differentiation from Existing Approaches

The proposed framework differentiates from existing approaches along four dimensions. First, bidirectionality: unlike the predominantly unidirectional BIM-optimization integrations documented by Machado et al. (2021), this framework establishes genuine two-way data flow between the GA engine and the BIM model, enabling the optimization to respond to BIM-computed material quantities rather than analytical approximations. Second, full code compliance: all twelve structural constraints are enforced within the GA fitness evaluation using explicitly implemented Eurocode 2 design equations, unlike surrogate or simplified approaches that may compromise code compliance for computational speed.

Third, accessibility: the framework employs entirely Python-based computation with widely accessible open-source libraries (NumPy, DEAP, SciPy) and widely deployed commercial BIM tools

(Revit, Dynamo), ensuring practical deployment in small and medium engineering practices without proprietary optimization software. This is particularly relevant to the Sub-Saharan African engineering context, where software accessibility and cost are significant practical constraints. Fourth, the explicit land development framing of the optimization objective: by formulating cost minimization in terms of material cost per structural element and demonstrating its aggregated impact on development economics, the framework provides decision-support that is directly interpretable by land developers, quantity surveyors, and project financiers, not only structural engineers.

4.3 GA Hyperparameter Justification

The selected GA hyperparameters are explicitly justified against benchmark literature values. The population size of N = 200 provides sufficient genetic diversity for global search while maintaining computational tractability within the BIM-integrated workflow, informed by the 240-280 individual populations recommended by Menezes et al. (2023) for comparable RC column problems. The crossover probability $P_c = 0.9$ follows the recommendation of Haupt and Haupt (2004) and is consistent with values used in beam and column optimization literature. The per-gene mutation probability $P_m = 1/n_{vars}$ (approximately 0.17-0.20) conforms to standard real-coded GA practice, with the 5% floor mutation rate adopted from Menezes et al. (2023).

The SBX distribution index $\eta_c = 20$ provides a balance between exploitation of parent values and exploration of adjacent regions, as validated by Deb and Agrawal (1995) for real-valued GA applications. The tournament size $k_t = 3$ delivers consistent selection pressure without accelerating premature convergence, important for problems with flat fitness landscapes near the optimum, as often encountered in constrained RC design. The penalty coefficient $\alpha = 10$ was calibrated to ensure infeasible solutions are clearly disadvantaged without completely eliminating them from the population, preserving diversity while directing the search toward feasible regions.

Table 1. Summary of GA Hyperparameters

Parameter	**Value**
Population size, N	200

Maximum generations	150
Crossover type	SBX (continuous); Uniform (discrete)
Crossover probability, Pc	0.9
SBX distribution index, ηc	20
Mutation type	Polynomial (continuous); Random reset (discrete)
Mutation probability, Pm	$1/n_{vars}$ (≈0.17–0.20)
Polynomial mutation index, ηm	20
Selection mechanism	Tournament selection (k = 3)
Elitism	Single best individual retained
Penalty coefficient, α	10.0
Independent runs	30 (global optimum verification)
Convergence tolerance	< 0.1% improvement over 30 generations

5. Conclusion

This paper has presented a unified, automated framework integrating Python-based Genetic Algorithm optimization with BIM parametric modelling for the systematic optimization of reinforced concrete beam and column sections under Eurocode 2. The framework addresses a well-documented gap in the literature: the absence of a transparent, bidirectionally integrated, fully code-compliant BIM-GA system applicable to practical RC structural design. Four key contributions have been established: a rigorous mixed-variable optimization formulation with eleven design variables and twelve structural constraints; a complete GA implementation with explicitly justified hyperparameters grounded in benchmark literature; a bidirectional BIM integration framework using Revit, Dynamo, and JSON-based data exchange; and a clear articulation of the framework's relevance to sustainable, cost-effective land development.

The land development framing of the optimization objective is a central contribution of this paper. By demonstrating that systematic structural material optimization directly reduces construction cost per unit of developed land and that this effect scales substantially across typical development portfolios, the framework provides decision-support that extends beyond the structural engineering discipline into development economics, quantity surveying, and infrastructure finance. In rapidly urbanizing emerging economies where construction cost is a primary constraint on land development viability, the practical value of this framework is particularly significant.

The exclusive use of Python and widely deployed commercial BIM tools ensures that the framework is not merely an academic artefact but a practically deployable solution accessible to small and medium engineering practices. Full quantitative results including convergence profiles, comparative cost analyses, and material efficiency assessments from the four-storey RC building case study will be reported in a subsequent publication upon completion of numerical experiments.

Future work will extend the framework to multi-storey irregular building forms, incorporate dynamic loading (wind and seismic), integrate lifecycle cost assessment, and explore surrogate modelling to reduce per-generation BIM update latency. The integration of machine learning-based surrogate models for structural performance prediction, replacing analytical EC2 constraint evaluations in computationally intensive scenarios, represents a particularly promising extension of the present framework.

References

Afzal, M., Li, R. Y. M., Ayyub, M. F., Shoaib, M., & Bilal, M. (2023). Towards BIM-based sustainable structural design optimization: A systematic review and industry perspective. Sustainability, 15(20), 15117. https://doi.org/10.3390/su152015117

Ahmad, S. A., Izhar, T., Ahmed, T., & Mumtaz, N. (2024). Transformative role of machine learning in design optimization of reinforced concrete frames. International Journal of Advanced Technology and Engineering Exploration, 11(112), 437–458. https://doi.org/10.19101/IJATEE.2023.10102336

Alsabbagh, A., & Abdulrahman, M. (2025). Design optimization of reinforced concrete slabs for sustainable construction (Report No. TVBK-25/5308). Division of Structural Engineering, Lund University.

Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. Leadership and Management in Engineering, 11(3), 241–252.

Benjamin, A. (2025). Structural engineering insights for improving building performance and ensuring long-term societal safety worldwide. QITP-IJSE, 5(2), 1–6.

Cabrera, P. A., Medina, G. M., & Delgadillo, R. M. (2025). Automation and genetic algorithm optimization for seismic modeling and analysis of tall RC buildings. Buildings, 15(19), 3618. https://doi.org/10.3390/buildings15193618

CEN. (2002). EN 1991-1-1: Eurocode 1: Actions on structures – Part 1-1: General actions – Densities, self-weight, imposed loads for buildings. European Committee for Standardization.

CEN. (2004). EN 1992-1-1: Eurocode 2: Design of concrete structures – Part 1-1: General rules and rules for buildings. European Committee for Standardization.

Coello, C. A. C., Christiansen, A. D., & Santos, F. (1997). Optimal design of reinforced concrete beams using genetic algorithms. Expert Systems with Applications, 12(1), 101–108.

Cruz, R. O., & Lemonge, A. C. C. (2020). Optimization of cross-sectional areas of reinforced concrete columns subjected to axial force and biaxial bending via genetic algorithms. Revista Mundi Engenharia, Tecnologia e Gestão, 5(2), 222-01–222-16.

Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. Complex Systems, 9(2), 115–148.

Eastman, C. M., Teicholz, P., Sacks, R., & Liston, K. (2011). BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors (2nd ed.). John Wiley & Sons.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

Haupt, R. L., & Haupt, S. E. (2004). Practical genetic algorithms (2nd ed.). John Wiley & Sons.

Holland, J. H. (1992). Adaptation in natural and artificial systems. MIT Press.

Machado, M., Scheer, S., & Santos, A. D. (2021). BIM-based structural optimization: A systematic review. Journal of Building Engineering, 33, 101835.

Menezes, I. S., Tinoco, V. N. V., Christoforo, A. L., Bomfim Junior, F. C., & Narques, T. V. N. (2023). Optimization of reinforced concrete columns via genetic algorithm. Acta Scientiarum Technology, 45, e61562.

Mitchell, M. (1996). An introduction to genetic algorithms. MIT Press.

Naser, M. Z., et al. (2022). Artificial intelligence and optimization in structural engineering design. Engineering Structures, 255, 113857.

Ranjan, N., & Singh, D. P. (2025). Development of a BIM-based framework for automated structural design optimization. International Journal of Research Publication and Reviews, 6(6), 7755–7763.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). BIM handbook (3rd ed.). John Wiley & Sons.

Sherif, M., Nassar, K., Hosny, O., Safar, S., & Abotaleb, I. (2022). Automated BIM-based structural design and cost optimization model for reinforced concrete buildings. Scientific Reports, 12, 21616.

Yang, X. S. (2010). Engineering optimization: An introduction with metaheuristic applications. John Wiley & Sons.

Woodbury, R. (2010). Elements of parametric design. Routledge.