ISSN 2729-7640

sciendo

ALFA
ARCHITECTURE PAPERS
OF THE FACULTY OF ARCHITECTURE AND DESIGN STU
www.alfa.stuba.sk

3/2025 (Vol. 30)
DOI: 10.2478/alfa-2025-0014

Innovative method for designing architectural plans and constructing prefabricated structural components using digital design solutions

Farnaz Rahimifard*

[1] *Islamic Azad University, Department of Architecture, Tehran, Iran*

***Corresponding author**
E-mail: farnazrahimifar@yahoo.com

Article information
Sent: Jun 3, 2025
Accepted: Jul 15, 2025
Published: Sep 12, 2025

Abstract: The growing complexity of urban development and the demand for faster, more efficient construction have led architects and engineers to explore new design paradigms. Among these, modular construction and prefabrication have emerged as promising solutions for high-rise buildings, offering advantages in speed, quality control, and material efficiency. Modular systems rely on the off-site production of standardised units, later assembled on-site using various installation techniques, reducing construction time and minimising waste. Parallel to this, advancements in digital design—especially parametric and algorithmic modelling—have transformed architectural workflows. These tools allow real-time simulation, spatial analysis, and iterative optimisation, enabling architects to create spatial configurations that are both functional and adaptable. Yet, a gap remains in integrating digital design logic with modular systems, particularly in high-rise residential architecture. This research addresses that gap by introducing a computational method for generating and optimising modular floor plans. The approach integrates algorithmic design, adjacency validation, and multi-objective optimisation within a parametric workflow. It supports rapid floor plan generation while responding to user preferences and functional requirements, site conditions, and architectural criteria. Unlike catalogue-based modular systems, the proposed method automates floor plan generation and evaluates performance across multiple objectives. It also offers scalability to larger urban contexts, contributing to more adaptable, sustainable construction. The aim of this study is to develop a computational design methodology that enables the generation, evaluation, and optimisation of modular floor plans using algorithmic logic and spatial constraints. It seeks to bridge the gap between conceptual digital design and practical implementation in high-rise modular architecture.

Keywords: digital architecture, algorithmic design, modular construction, prefabricated components, architectural planning

INTRODUCTION

With the rapid growth of urban populations and the consequent increase in land prices, the construction of tall buildings has become increasingly prevalent. Due to the significant consumption of materials, energy, and labour in such projects, economic and environmental efficiency have become key concerns. As a result, architects and engineers continue to explore innovative strategies that reduce costs and carbon footprints while maintaining construction quality and design integrity.

Contemporary high-rise construction employs a range of methods, each with its own distinct characteristics. Among these, modular construction using prefabricated components has emerged as a promising approach. It enhances fabrication speed, minimises material waste, and improves on-site efficiency (Knaack et al., 2012). Modular buildings are composed of prefabricated units manufactured off-site, customised according to the functional requirements of the structure. After passing quality assurance tests, the modules are transported and assembled using various installation techniques, significantly reducing construction time (Lawson et al., 2014).

The integration of digital technologies in architecture and engineering has profoundly transformed both the design and construction processes. From early conceptualisation to implementation, computer-aided tools—especially algorithmic and parametric design techniques—enable higher precision, optimisation, and flexibility. Recent research confirms that algorithmic approaches are particularly effective in the design of modular systems, allowing for spatial adaptability and enhanced performance metrics (Oxman, 2017).

Despite these advancements, a clear and systematic methodology that fully integrates digital tools with prefabricated modular construction in high-rise architecture is still lacking. This gap motivates the current study. This research introduces an innovative method for designing architectural floor plans and constructing prefabricated structural components using algorithmic design tools. It focuses on high-rise residential buildings and demonstrates how digital workflows can improve spatial efficiency, reduce construction time, and support more sustainable urban development. It is hypothesised that applying parametric strategies within modular design can yield optimised spatial configurations that are both cost-effective and architecturally responsive.

© 2025 Author. This is an open access article licensed under the Creative Commons Attribution-International License (https://creativecommons.org/licenses/by/4.0/). Published by the Faculty of Architecture and Design, Slovak University of Technology in Bratislava, Slovakia

DIGITAL COMPUTATION

Digital technologies have deeply permeated every stage of architectural practice—from the early stages of conceptual design to the management of building information and, ultimately, to on-site fabrication. Today, architects harness computational power to engage with the complex characteristics of materials and transform them into generative agents for design. Contemporary digital design approaches—often referred to as digital architectures—are grounded in computational theories and include concepts such as topological spaces (topological architecture), isomorphic surfaces, motion dynamics (animate architecture), morphing forms (metamorphic architecture), parametric modeling, and genetic algorithms (evolutionary architecture).

The effectiveness of digital architecture in generating meaningful and context-aware designs is closely tied to the perceptual and cognitive abilities of the designer. These processes are inherently dynamic and require qualitative reasoning to guide the emergence of form. Once components are digitally designed, their fabrication and on-site assembly can be supported by digital tools. Three-dimensional models, for example, can be used to determine component positioning, guide robotic or manual placement, and ensure accurate construction.

By integrating the design, analysis, manufacturing, and assembly phases into a digitally coordinated workflow, architects and engineers can revive the integrated "master-builder" role—reuniting disciplines that had become fragmented in the post-industrial era. As noted by Mitchell and McCullough (1995), digital workflows help close the long-standing gap between architectural design and physical construction. At its core, digital computation in architecture involves the use of algorithms, codes, and rules within software environments to solve design problems. This process transforms raw input data into optimised design solutions through rule-based modelling, simulation, and evaluation.

While computational methods can be used for a wide range of applications—from statistical analysis to search and automation—in the context of architecture, this paradigm is known as Design Computation. It goes beyond simply using computers for drafting; it encompasses an entire methodology of problem-solving informed by computational logic and geometry. Parametric design and algorithmic modelling fall under this umbrella, offering architects the ability to create adaptable, efficient, and intelligent modular design systems that respond to contextual inputs in real time.

COMPUTATIONAL ARCHITECTURE

As computational technologies become increasingly embedded in daily life, their influence has extended deeply into the architectural domain. Computational architecture, as a subfield, is rapidly evolving—attracting growing interest from designers, researchers, and engineers. Information and communication technologies (ICT), particularly digital design and fabrication tools, have become integral components of architectural practice, reshaping both design methodology and construction techniques. Over the past three decades, the convergence of computation and architecture has fostered a transformation in how buildings are conceptualised, analysed, and realised. Computational architecture serves as a critical bridge between the tangible world of physical structures and the abstract realm of digital information. It mediates between matter—the heaviest and most concrete aspect of the built environment—and data—the most intangible and virtual.

This dynamic interaction enables architects to synthesise geometric logic, material behaviour, and environmental data into cohesive design strategies. As Khabbazi (2022) explains, computational architecture reinforces this physical-digital relationship, enabling more intelligent, adaptive, and efficient design solutions. As noted by Gary William Flake (1998), computational systems operate by processing discrete inputs over defined intervals, transforming them into meaningful outputs. These systems function based on strings of input or output interactions, enabling them to simulate complex systems through algorithmic control. In architectural design, this logic supports the generation of adaptive spatial layouts, responsive forms, and optimised structural layouts. By using advanced computational frameworks, designers can manage complex constraints, explore large design spaces, and produce innovative spatial solutions beyond the reach of traditional design approaches.

FABRICATION-AWARE DESIGN

Fabrication-aware design is a progressive design paradigm that actively integrates the possibilities and constraints of digital fabrication processes into the architectural workflow. With the widespread adoption of CNC (Computer Numerical Control) technologies—such as laser cutters, plasma cutters, thermal and water jet cutting machines, milling devices, and robotic arms—the role of fabrication has shifted from a post-design phase to a driving factor in shaping architectural forms. Each digital fabrication tool introduces specific parameters and limitations that influence material behaviour, form generation, and detailing. A fabrication-aware designer not only understands these constraints but strategically leverages them to enhance architectural expression and constructability.

This design approach recognises that fabrication is not a passive act of translating drawings into physical objects, but rather an active, formative process that contributes to the identity and logic of the design itself. It requires a deep understanding of machine behaviour, material properties, tolerances, and sequencing strategies. As KhalilBeigi (2023) emphasises, the unintended effects of digital tools can become sources of innovation when designers engage with them critically. By anticipating how materials will respond during the fabrication process, and by embedding this intelligence into the design from the outset, architects can create solutions that are not only efficient and precise but also rich in performative and aesthetic value.

Ultimately, fabrication-aware design facilitates a seamless integration between digital conception and material realisation. It transforms the design process into a looped feedback system, where design intentions are tested, refined, and implemented based on the realities of production—resulting in architecture that is both digitally intelligent and materially grounded.

MODULAR DESIGN

Modular design, often referred to as modularity in design, is a systematic approach that divides a complex architectural system into discrete, standardised units or modules. These modules can be independently created, modified, replaced, or reused across various contexts without altering the integrity of the entire system. This principle enhances design flexibility, scalability, and efficiency in both planning and construction. Each modular unit is developed based on a defined dimensional system, ensuring interoperability and consistency. The overall architectural form evolves through the composition and extension of these modules. Importantly, changes to one module do not necessitate alterations in others, allowing localised modifications without compromising the system as a whole (Baldwin, Clark, 2015).

A modular system typically features:

- Functional partitioning into reusable units,
- Clearly defined interfaces between modules,
- Adherence to industry standards for connectivity and assembly.

Modular systems enhance adaptability and sustainability by allowing seamless integration of new components and easy replacement of obsolete parts with minimal disruption to the overall structure. Furthermore, modularity enhances the capacity for prefabrication, allowing components to be manufactured in controlled factory environments and assembled on-site with precision. This reduces labour time, environmental impact, and errors during construction. In the context of high-rise or urban construction, modular design proves especially beneficial by minimising on-site activities, mitigating disturbances to neighbouring buildings, and supporting phased or vertical expansions. It aligns effectively with computational workflows and digital fabrication methods, supporting modular spatial system optimisation.

CASE STUDIES

To contextualise the theoretical framework of this research, several case studies are examined that illustrate the application of modular and prefabricated strategies in architectural design and construction. These examples highlight the potential of modularity in high-rise buildings and demonstrate how digital tools and prefabrication techniques are being leveraged in practice.

Nakagin Capsule Tower, Tokyo, Japan (1972)

One of the earliest examples of modular high-rise architecture, the Nakagin Capsule Tower was designed by Kisho Kurokawa and constructed in the Ginza district of Tokyo. The 14-storey building comprises 140 prefabricated capsules, each rotated around a central core and attached using only four high-tension bolts. This enabled each capsule to be independently installed, replaced, or removed with relative ease (Fig. 1). Each capsule, measuring approximately 4 × 2.5 metres, was factory-fabricated—including all interior components such as a bed, bathroom, storage, and electronic equipment—and transported to the site for installation. The design allowed for capsule combinations to create larger living spaces by connecting adjacent units. Despite its visionary concept, the project faced time constraints, with design continuing even after construction began (Sveiven, 2011). This project stands as a significant early reference for prefabricated modular design in dense urban environments.

Although Nakagin Capsule Tower pioneered prefabricated modularity, it suffered from rigid spatial standardisation and lack of long-term adaptability. The fixed structural integration of capsules with the core made modular rearrangement or typological diversity nearly impossible after initial installation. This limitation highlights the need for dynamic, data-driven planning tools capable of evaluating multiple floor plan layout configurations before implementation. In this study, the proposed algorithmic framework addresses this issue by enabling the generation, filtering, and optimisation of plan arrangements using parametric design and SPEA-2 algorithm, which were not accessible during Nakagin's era.

The overall footprint of the tower was compact, with each floor housing 10 capsules organised around a central core. The core included vertical circulation shafts and mechanical systems, while the surrounding modules were cantilevered outward. The building did not include a traditional podium; instead, the tower rose directly from a small base structure with limited public or shared space. This configuration emphasised vertical modular stacking over horizontal public integration, reflecting the high-density, land-constrained context of central Tokyo.

While Nakagin Capsule Tower represents an early example, more recent explorations into modular capsule living include the Urban Village Project developed by SPACE10 and IKEA in 2019 (SPACE10, 2019). Unlike the rigid capsule-core structure of Nakagin, the Urban Village system proposes highly flexible, stackable living modules supported by digital customisation and community-based services. These newer models highlight the growing relevance of adaptability and smart integration—values which are also central to the computational framework proposed in this study.

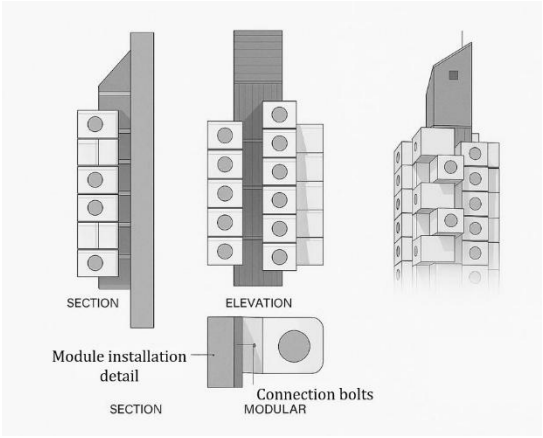

Fig. 1. Section, elevation, and a reference photo of the Nakagin Capsule Tower, illustrating the modular spatial units and their connection details to the central core. (Source: Author, 2025)

Penda Modular Tower, Vijayawada, India (2015)

The Penda Modular Tower represents a more contemporary interpretation of modular design, emphasising user participation and environmental responsiveness. The system enables residents to customise their apartments by selecting prefabricated modules from a catalogue. These modules are then inserted into a preconstructed structural framework (Fig. 2).

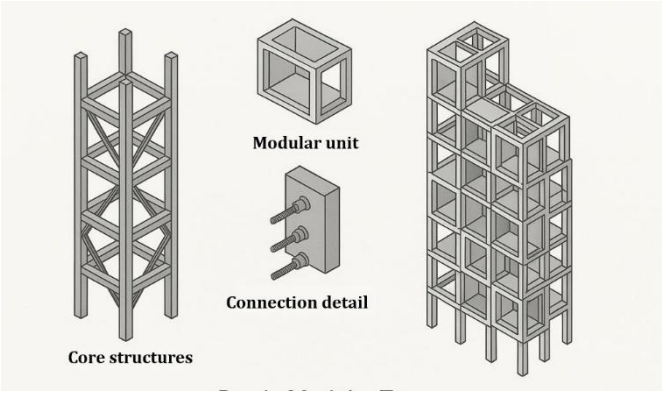

Fig. 2. Modular structural components of the proposed Penda Tower, demonstrating the assembly logic and vertical stacking of prefabricated units. (Source: Author, 2025)

The building consists of eight prefabricated components, including the structural grid system, walls, façade, floor slabs, infrastructure, balconies, and planters. This "kit-of-parts" strategy empowers residents to co-design their living environments, fostering personalisation within a standardised system. The tower's open corridors and varied façades enhance natural ventilation and daylight access, while integrated systems support water reuse and vertical greenery. Penda refers to the project as a "breathing tower," highlighting its ecological integration and architectural flexibility (Rosenfield, 2025).

While Penda's system advances flexibility and user participation, it lacks algorithmic rigor in determining optimal unit configurations. The selection of modules is primarily catalogue-based and subjective, rather than generated or filtered computationally. By contrast, the proposed methodology in this research formalises the design process through a computational framework that automatically generates invalid spatial configurations and identifies optimal spatial configurations based on architectural, func-

tional, and geometric criteria. In this sense, Penda's vision is conceptually aligned with the goals of this study, but the current work introduces a more systematic and scalable digital design logic.

The conceptual design proposes a tower of approximately 20–25 storeys, though actual implementation details are not finalised. Each modular apartment occupies a floor plan area of roughly 60–80 square metres, depending on the module selection. The podium spans three storeys and serves as a shared public base including amenities and infrastructure services. The structural grid follows a 4.5-metre spacing in both directions to support flexible module insertion. Although the project remains unbuilt, these proposed dimensions offer insight into the potential scalability and adaptability of the modular system.

Modular system research by Scott Howe (2000)

In a foundational study published in Automation in Construction, Scott Howe proposed a modular building system oriented toward automated construction. His work identifies four primary spatial types—user space, circulation space, core or service space, and exterior space—and explores how their dimensions and configurations can be adapted within a modular framework. Howe's model introduces a "component grammar" based on a basic spatial grid, which defines both structure and spatial boundaries. Modules are designed with standard widths while allowing flexibility in depth, enabling a wide range of configurations. When combined, adjacent structural zones may overlap or remain independent, supporting parallel assembly and construction automation (Fig. 3).

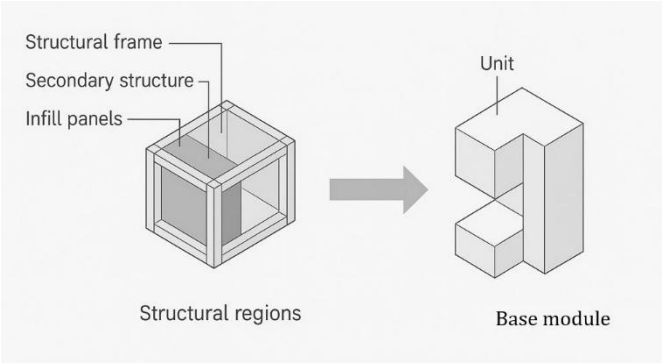

Fig. 3. Structural regions of the prefabricated modules (left) and the geometric shaping of spatial units based on the base module (right). (Source: Author, 2025)

A key insight of this system is its scalability and adaptability for different user preferences and functional requirements, including educational, commercial, or residential functions. Enabled by a grammar-based approach, the system supports automated decision-making, allowing for the generation of flexible, efficient, and cost-effective modular configurations. In the design implementation, the plan was conceived as a linear arrangement with several bays placed side by side, following the rules of the shape grammar. Functional areas were separated by movable partitions, with bay widths varying according to spatial requirements. Core facilities such as restrooms, kitchen, offices, and the library were concentrated in designated modules, thereby maximising the openness and flexibility of the surrounding spaces for teaching and training purposes (Fig. 4).

Howe's component grammar introduces a foundational logic for modular planning, but its manual application limits responsiveness to large-scale urban constraints. The optimisation framework in this study builds on such grammar-based systems but extends them through algorithmic decision-making and performance evaluation using evolutionary computation. The use of adjacency rules and geometric filtering integrates spatial logic with architectural feasibility, offering an advancement over static grammar systems by incorporating flexibility, responsiveness, and automation. In Howe's prototype, each bay was approximately 3.6 metres wide, with varying depths based on function. Circulation modules maintained a consistent depth of 1.5 metres, while user zones ranged from 3 to 5 metres in depth. These dimensions were chosen to reflect human-scale usability and prefabrication logistics. The modules were arranged linearly along a structural spine, forming a flexible and reconfigurable teaching environment.

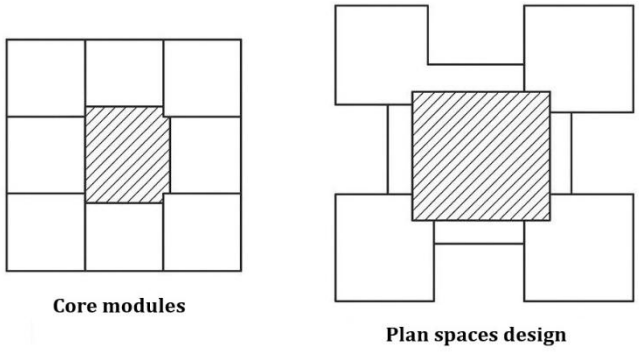

Fig. 4. Core modular spatial units (left) and the corresponding plan spaces design (right) illustrating spatial arrangement based on modular logic (used in the Scott Howe modular study). (Source: Author, 2025)

These case studies underscore the practical relevance of modular strategies but also reveal limitations in adaptability, customisation, and performance-driven decision-making. Their analysis directly informed the development of the proposed method, shaping the algorithm's focus on spatial flexibility and data-driven layout optimisation. In response, the following section introduces a computational methodology designed to address these challenges through parametric modeling and multi-objective evaluation.

METHODOLOGY

This research proposes a multi-stage methodological framework for generating, evaluating, and optimising architectural floor plan layouts using modular and computational design techniques. The approach integrates geometric operations, algorithmic filtering, and multi-objective optimisation to identify the most efficient layout alternatives within a modular design system.

Generation of probable floor plan layout

The first step involves generating a comprehensive set of possible spatial arrangements—referred to as the design space. This is achieved through simple geometric transformations such as translations and rotations applied to basic spatial units or "micro spaces." These units, represented as basic geometric shapes, are positioned adjacent to one another in varying sequences to simulate potential plan compositions (Fig. 5). To ensure architectural coherence, the algorithm simulates spatial arrangements not only by treating each micro space as an independent entity but also by aggregating them into cohesive, unified layouts. This ensures that the generated plans meet the minimum geometric and spatial continuity required for architectural feasibility.

The algorithm is primarily suited for high-rise residential buildings ranging from 15 to 40 storeys, where modular repetition and spatial efficiency are critical. However, the computational logic is adaptable to both tower block (point block) and slab block configurations. While the case application in this study focuses on a compact, vertical core typology, the layout generation me-

chanism can be scaled to larger footprints or horizontally extended arrangements by modifying boundary constraints and adjacency inputs. This flexibility allows for contextual adaptation in dense urban or linear site conditions.

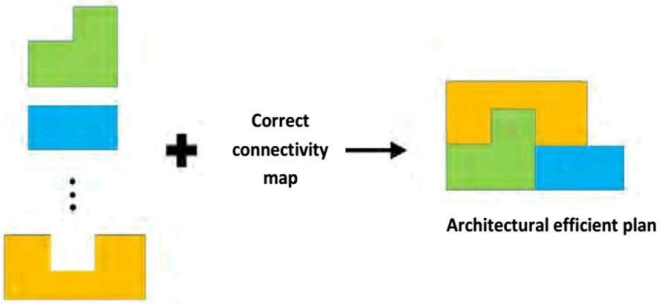

Fig. 5. Schematic representation of the initial generation of the design space, illustrating the architecturally efficient plan through transformation of basic shapes by translation and rotation. (Source: Author, 2025)

The current version of the algorithm is optimised for rectilinear and grid-based spatial arrangements, where modular spatial units can be composed using edge-to-edge logic within a Cartesian coordinate system. While it is theoretically possible to extend the approach to irregular forms such as circular, elliptical, or polygonal (e.g., octagonal) plan geometries, doing so would require significant adaptation of the adjacency rules and spatial validation routines. Future versions of the system may incorporate enhanced shape recognition algorithms and flexible boundary constraints to support non-orthogonal planning environments.

Filtering invalid spatial configurations

Due to the vast number of combinations, many layout alternatives may be architecturally invalid—resulting from spatial overlaps, lack of integration, or violation of desired adjacency relationships. Therefore, a refinement mechanism is employed to filter out inappropriate configurations before entering the optimisation phase. Instead of relying solely on computationally expensive geometric analysis, the filtering process employs a text-based method in which each spatial unit is assigned a unique character (e.g., A, B, C), and the algorithm determines the validity of a layout by comparing the sequence of these spatial units against an adjacency map; this spatial connectivity and adjacency rules are visualised in Fig. 6. and the user-defined adjacency requirements are encoded as strings (e.g., ABBC). The algorithm checks each layout alternative against predefined spatial adjacency rules, as illustrated in Fig. 7.

This text-based method offers several advantages:

- It ensures adjacency through shared edges rather than mere points, aligning with architectural standards.
- It significantly reduces computational load compared to geometric methods.
- It guarantees that undesired adjacencies are systematically excluded.
- It can be extended to incorporate environmental, structural, or climatic constraints by modifying the input string rules (Fig. 7, 8).

These methods were selected for their ability to handle the complexity and scale of modular high-rise layouts. The string-based adjacency filter significantly reduces computational overhead compared to geometric analysis, while SPEA-2 provides robust multi-objective optimisation with proven efficiency. Together, they form a streamlined process suitable for architectural scenarios where rapid feedback and adaptability are essential.

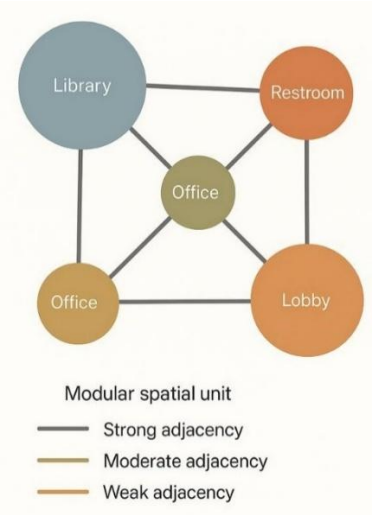

Fig. 6. Bubble diagram visualising spatial connectivity and adjacency logic used in the layout validation phase. Each bubble represents a micro-space, with lines indicating edge-to-edge adjacency based on user-defined rules. (Source: Author, 2025)

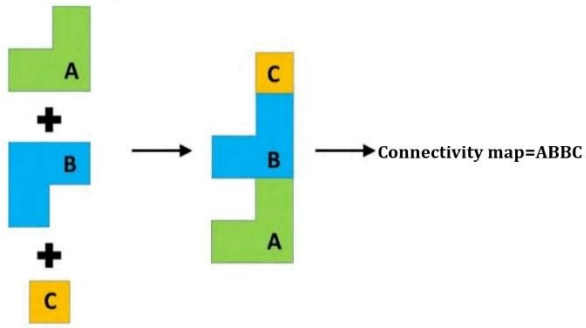

Fig. 7. Extracting the connectivity map from user input as a text string, ensuring edge-to-edge adjacency, systematic elimination of undesired adjacencies, and efficient extensibility based on various constraints. (Source: Author, 2025)

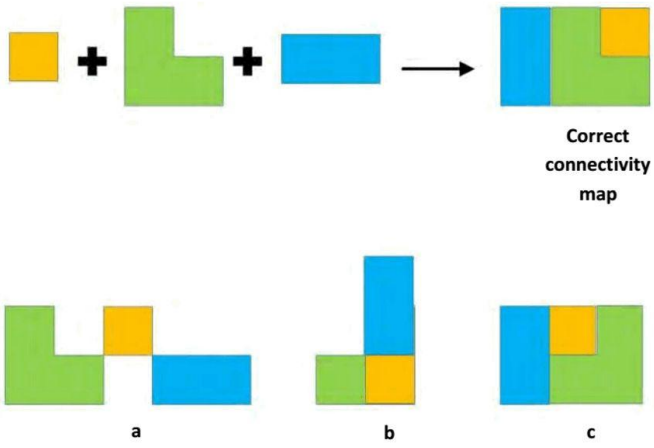

Fig. 8. Examples of unacceptable options in design space: a) unacceptable due to lack of integrity, b) unacceptable due to intersection and overlap of the micro spaces, c) unacceptable due to lack of correct comply from intended connectivity map. (Source: Author, 2025)

Optimisation using the SPEA-2 algorithm

Once the invalid spatial configurations are identified, a multi-objective optimisation algorithm process is implemented using the Strength Pareto Evolutionary Algorithm 2 (SPEA-2). This algorithm is selected for its ability to:

- Handle multi-criteria objectives,
- Filter out invalid inputs prior to optimisation,
- Deliver efficient results with lower computational cost compared to other evolutionary methods.

Alternative evolutionary algorithms such as NSGA-II, Genetic Algorithms (GA), and Particle Swarm Optimisation (PSO) were also reviewed. However, SPEA-2 was chosen due to its stronger elitism mechanism, lower computational overhead, and proven performance in spatial layout optimisation tasks where non-dominated solution sets are desired. These qualities make it particularly suitable for modular architectural applications that require balancing compactness, adjacency, and functionality.

In this study, the primary optimisation criterion is the total area of the floor plan. The objective is to minimise spatial footprint while preserving the desired adjacency and functionality. The most compressed (space-efficient) configuration is considered the optimal solution. To implement the optimisation process, the Octopus plugin was utilised as a multi-objective evolutionary tool within the parametric design environment. Octopus enables the integration of algorithms like SPEA-2 and facilitates the exploration of diverse design solutions by simultaneously evaluating multiple performance criteria (Vierlinger, 2017).

To clarify the multi-objective optimisation process, the following pseudocode outlines the complete algorithm used to generate and evaluate spatial configurations using the SPEA-2 method:

Algorithm: Modular Floor Plan Optimisation

Input: Set of micro-spaces and adjacency rules

Output: Optimal spatial configuration

1. Generate initial spatial units (micro-spaces)
2. Apply geometric transformations (translation, rotation)
3. Assemble multiple floor plan layouts
4. For each layout: a. Convert layout to string (e.g., ABBC) b. Check adjacency constraints c. If invalid → discard d. If valid → store in valid set
5. Run SPEA-2 Algorithm applied to the set of valid spatial configurations a: Evaluate objective function (e.g., area, compactness) b: Select non-dominated solutions
6. Return best configuration (Fig. 9).

The optimisation was performed using the Octopus plugin in Grasshopper. Each iteration for a mid-sized floor plan (10–12 micro-spaces) took approximately 4–6 seconds, and the complete optimisation cycle converged in under 8 minutes. The algorithm is scalable and can handle large solution spaces by adjusting population size and generation limits. In addition to reducing spatial footprint through optimisation, the method enables the early identification of material-efficient configurations. Although direct cost estimations are not included in this study, minimising plan area while maintaining functionality suggests potential savings in construction materials and labour. However, the methodology relies on access to computational design tools and digital workflows, which may limit its applicability in low-tech or resource-constrained environments.

In addition to the Octopus plugin used within the Grasshopper environment, the proposed algorithmic approach is also conceptually compatible with other Building Information Modeling (BIM) platforms such as Autodesk Revit, particularly when used in conjunction with visual programming tools like Dynamo. These environments support parametric rule-based modeling and can facilitate the integration of adjacency rules, optimisation logic, and spatial validation routines. Therefore, the proposed system could potentially be adapted or re-implemented within BIM-based workflows to enhance coordination between architectural design and construction documentation. Future development could address this by creating simplified interfaces or integrating with open-source platforms for broader use.

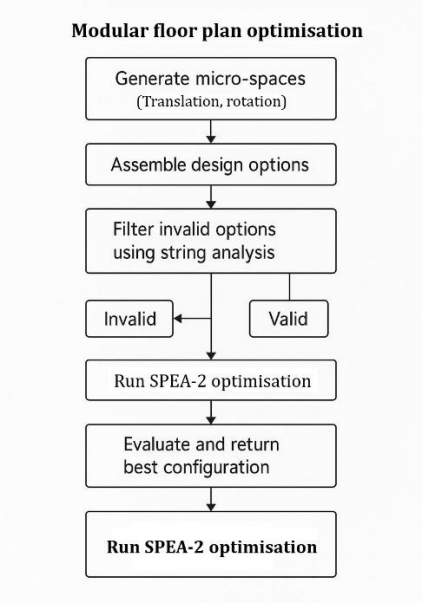

Fig. 9. Flowchart of the modular floor plan optimisation process using the SPEA-2 algorithm. (Source: Author, 2025)

FUTURE DEVELOPMENTS

Building upon the algorithmic framework established in this study, several directions for further development are proposed to enhance the adaptability, functionality, and contextual responsiveness of the design system.

Regional constraints and zoning integration

One key enhancement is the incorporation of spatial zoning within the algorithm's input. By assigning fixed spatial zones (e.g., north, south, east, west), users can guide the location of specific micro spaces relative to environmental or functional requirements. For instance, service cores can be constrained to the central or southern zones, while living spaces can be oriented toward sunlight-rich directions. This would improve both environmental performance and user satisfaction. The same text-string mechanism used for adjacency validation can be adapted to encode zone-specific placement rules. This ensures that adjacency relationships are maintained not only among internal spaces but also with respect to broader site or climatic contexts.

Functional integration of building equipment

Another area for expansion involves the integration of building systems and equipment into the modular layout. A complementary algorithm could be introduced to allocate technical infrastructure such as HVAC shafts, electrical conduits, or plumbing lines. By embedding this layer early in the plan generation process, the system can produce more construction-ready solutions, reducing the need for post-design modifications.

Structural optimisation layer

In addition to spatial layout, structural logic can also be embedded into the design framework. By defining structural zones—such as column grids, core walls, or load-bearing partitions—as fixed regions within the layout, the algorithm can account for structural feasibility while proposing spatial layouts. This would allow for the simultaneous optimisation of both spatial and structural systems, ensuring material efficiency and structural integrity. Structural constraints can be assessed iteratively during layout generation and refinement stages to identify the most robust floor plan layout. These potential enhancements would increase the intelligence, flexibility, and real-world applicability of the proposed system, making it suitable for a wide range of building types and design scenarios. The ultimate goal is to develop an integrated, multi-criteria design engine that merges spatial logic, structural reasoning, and environmental responsiveness within a unified computational platform.

APPLICATION OF THE ALGORITHM

To demonstrate the practical utility of the proposed algorithm, it has been applied to the design of a modular residential tower. In this scenario, each floor of the tower is composed of prefabricated modules, designed with specific dimensions to allow off-site fabrication and on-site assembly. This approach aims to accelerate construction timelines while ensuring high-quality execution and reducing environmental disruption. Based on the spatial requirements of a typical one-bedroom apartment, the modular spatial units are combined to form distinct functional zones. The modules are categorised and grouped as follows:

- Single-module spaces: Entrance, WC, bathroom, storage, and terrace
- Three-module space: Corridor
- Four-module spaces: Living room, kitchen, dining area, and bedroom

Each module is conceived as a square unit, providing geometric uniformity and allowing various combinations for spatial planning (Fig. 10). The spatial layout is defined by a user-provided pattern indicating which modules are to be combined and how they should relate to one another. The algorithm processes this input and begins generating multiple plan alternatives by arranging the modular spatial units within a square bounding frame. It evaluates each proposed layout based on spatial efficiency, focusing particularly on minimising unused or residual space (Fig. 11). The user-defined spatial relationships serve as constraints, ensuring that the generated plans maintain architectural logic and functional adjacency. The algorithm ranks each alternative and offers optimised options, from which the architect can select the most appropriate solution based on qualitative or contextual criteria (Fig. 12).

This process exemplifies the symbiosis between algorithmic automation and human judgment: while the system generates and evaluates numerous options, the architect retains the creative agency to interpret, refine, or even reject them. This interaction redefines the role of the architect within computational design—not as a passive consumer of digital outcomes, but as an informed co-author working alongside intelligent systems. The outcomes of this implementation confirm the feasibility of the proposed algorithm in generating adaptable and efficient modular layouts. The following conclusion synthesises the research findings and highlights the broader implications for architectural practice and education.

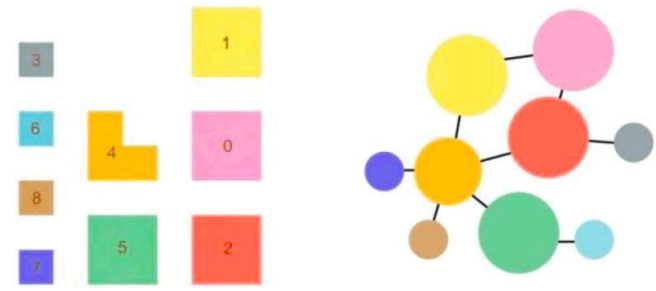

Fig. 10. Left: Modular spatial units conceived as square geometries to ensure uniformity and enable flexible spatial combinations. Right: Connectivity relations between spaces based on a user-defined pattern specifying how modules are combined and spatially related. (Source: Author, 2025)

Fig. 11. Left: modular spatial units; Middle: example of a plan with correct space relations (defined by the user); Right: placing the pieces in a square bounding frame. The bounding square represents a 10 × 10 metre area, and each module is 2.5 × 2.5 metres. Generated using Grasshopper (Rhino) during algorithmic layout. (Source: Author, 2025)

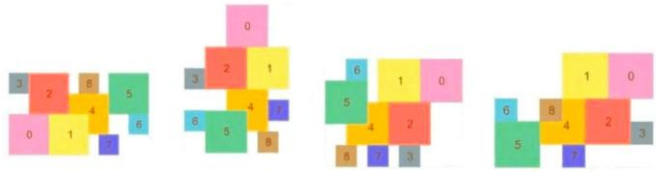

Fig. 12. Alternative modular floor plan layouts proposed by the algorithm, based on spatial constraints defined by the user. Each layout is generated using Grasshopper (Rhino), and the total bounding area is approximately 10 × 10 metres, with each module measuring 2.5 × 2.5 metres. (Source: Author, 2025)

CONCLUSION

This research presents a comprehensive methodology for integrating modular design and computational architecture to optimise the planning and construction of high-rise buildings. By leveraging algorithmic tools and digital fabrication strategies, the proposed approach enables the generation of architectural layouts that are not only efficient and adaptable but also structurally and environmentally responsive. The study demonstrates how computational methods—specifically algorithmic plan generation, adjacency validation through text-based string analysis, and multi-objective optimisation—can be applied to create functional, prefabricated spatial layouts. These methods enhance both the design process and construction workflow, resulting in reduced material waste, faster project delivery, and minimised on-site disruption.

Although the case study focused on residential units, the underlying principles and modular framework can be extended to various building types and functions. The flexibility of the system allows designers to tailor configurations to user preferences and functional requirements, as well as site-specific constraints. Furthermore, by combining modularity with digital fabrication, the approach supports mass customisation and industrialised building processes—key requirements for sustainable urban development in densely populated areas.

Ultimately, this study reaffirms the potential of computational design to bridge the gap between architectural creativity and constructability. By embedding intelligence into the design process and optimising spatial relationships from the outset, architects can produce solutions that are simultaneously innovative, feasible, and future-ready. Beyond its application in practice, the proposed method holds pedagogical value for architectural education. By introducing students to algorithmic reasoning, parametric design, and performance-based evaluation, this framework can help bridge the gap between conceptual design thinking and data-driven spatial logic. Embedding such computational strategies into studio education may empower emerging designers to develop solutions that are both creatively expressive and technically optimised.

References

Baldwin, C.Y., Clark, K.B. (2000) 'Design Rules, Volume 1: The Power of Modularity', MIT Press, Cambridge, USA. https://doi.org/10.7551/mitpress/2366.001.0001

Flake, G.W. (1998) 'The computational beauty of nature: Computer explorations of fractals, chaos, complex systems, and adaptation', MIT Press, Cambridge, USA.

Howe, S. (2000) 'Designing for automated construction', Automation in Construction, 9(3), pp. 259–276. https://doi.org/10.1016/S0926-5805(99)00041-2

Khabbazi, Z. (2022) 'Digital design processes', Kasra Bookcase, Tehran, Iran. [in Persian]

KhalilBeigi, A. (2023) 'Form-finding and optimization of funicular netted shells; designing a science and technology exhibition', master's thesis, Islamic Azad University, Tehran, Iran, Islamic Azad University Repository. [in Persian]

Knaack, U., Chung-Klatte, S., Hasselbach, R. (2012) 'Prefabricated systems: Principles of construction', Birkhäuser, Basel, Switzerland.

Lawson, M., Ogden, R., Goodier, C. (2014) 'Design in modular construction', CRC Press, London, UK. https://doi.org/10.1201/b16607

Mitchell, W.J., McCullough, M. (1995) 'Prototyping', In: Digital design media (2nd ed.), Van Nostrand Reinhold, New York, USA, pp. 417–440.

Oxman, R. (2017) 'Thinking difference: Theories and models of parametric design thinking', Design Studies, Vol. 52, pp. 4–39. https://doi.org/10.1016/j.destud.2017.06.001

Rosenfield, K. (2025) 'Penda to build modular customizable housing tower in India', ArchDaily. [online] Available at: https://www.archdaily.com/772181/penda-to-build-modular-customizable-housing-tower-in-india [Accessed: 8 Sep 2016]

SPACE10 (2019) 'The Urban Village Project: Re-envisioning Our Homes', SPACE10. [online] Available at: https://space10.com/project/urban-village-project

Sveiven, M. (2011) 'AD Classics: Nakagin Capsule Tower / Kisho Kurokawa', ArchDaily. [online] Available at: https://www.archdaily.com/110745/ad-classics-nakagin-capsule-tower-kisho-kurokawa/ [Accessed: 8 Sep 2016]

Vierlinger, R. (2017) 'Octopus' (multi-objective optimisation plugin), food4Rhino – Apps for Rhino and Grasshopper. [online] Available at: http://www.food4rhino.com/app/octopus/ [Accessed: 9 Apr 2017]