ICCEPM 2024 https://dx.doi.org/10.6106/ICCEPM.2024.0585
The 10th International Conference on Construction Engineering and Project Management
Jul. 29-Aug.1, 2024, Sapporo

Multi-objective Generative Design Based on Outdoor Environmental Factors: An Educational Complex Design Case Study

Kamyar FATEMIFAR[1*], Qinghao ZENG[2], Ali TAYEFEH-YARAGHBAFHA[3], Pardis PISHDAD[4]

[1] *School of Building Construction, Ph.D. student, Georgia Institute of Technology, the United States,* E-mail address: kfatemifar3@gatech.edu
[2] *School of Building Construction, Ph.D. student, Georgia Institute of Technology, United States,* E-mail address: qzeng41@gatech.edu
[3] *School of Architecture, B.A., IAUCTB, Iran,* E-mail address: ality2332@gmail.com
[4] *School of Building Construction, Associate Professor, Georgia Institute of Technology, United States,* E-mail address: pardis.pishdad@design.gatech.edu

Abstract: In recent years, the construction industry has rapidly adopted offsite-manufacturing and distributed construction methods. This change brings a variety of challenges requiring innovative solutions, such as the utilization of AI-driven and generative design. Numerous studies have explored the concept of multi-objective generative design with genetic algorithms in construction. However, this paper highlights the challenges and proposes a solution for combining generative design with distributed construction to address the need for agility in design. To achieve this goal, the research delves into the development of a multi-objective generative design optimization using a weighted genetic algorithm based on simulated annealing. The specific design case adopted is an educational complex. The proposed process strives for scalable economic viability, environmental comfort, and operational efficiency by optimizing modular configurations of architectural spaces, facilitating affordable, scalable, and optimized construction. Rhino-Grasshopper and Galapagos design tools are used to create a virtual environment capable of generating architectural configurations within defined boundaries. Optimization factors include adherence to urban regulations, acoustic comfort, and sunlight exposure. A normalized scoring approach is also presented to prioritize design preferences, enabling systematic and data-driven design decision-making. Building Information Modeling (BIM) tools are also used to transform the optimization results into tangible architectural elements and visualize the outcome. The resulting process contributes both to practice and academia. Practitioners in AEC industry could gain benefit through adopting and adapting its features with the unique characteristics of various construction projects while educators and future researchers can modify and enhance this process based on new requirements.

Key words: generative design, genetic algorithm, educational complex, Rhino-Grasshopper optimization

1. INTRODUCTION

In the urge of construction industrialization, there is a significant trend towards distributed manufacturing and modular construction within the Architecture, Engineering, and Construction (AEC) sector[1], [2]. With the aim of proposing sustainable and affordable housing solutions, along with its foreseeable economic boon, a variety of challenges are created, requiring innovative methodologies [3].

In order to actualize the idea of productizing the building components and advance modular construction, efficient and flexible design solutions for building components and modules are of critical necessity [4], which underscores the potential assistance with Artificial Intelligence (AI) and machine learning algorithms. While AI originated applications are gaining their position in the AEC industry [5], generative design has experienced a significant rise in interest and innovation [6]. Within the dynamic framework of industrial production, numerous decisions should be made by designers and architects within a limited timeframe [7]. Consequently, in order to facilitate the decision-making process and achieve optimized results for various specifications, generative design is applied not only to ensure the building quality but also the adherence to sustainable construction principles. As one of the essential tools for generative design, genetic algorithms are entering a seminal phase where researchers target at leveraging design optimization for solving complex and intricate design challenges with efficient and feasible computing power [8].

In this paper, one specific case utilizing multi objective design optimization is proposed and analyzed. Our goal is to design an aesthetically appealing, economic, scalable, and consequently modular educational and residential complex which follows the urban regulations and provides a well-lit, quiet, and accessible environment for occupants. The building is designed to foster the development of talented individuals with financial limitations. On the one hand, the non-profit nature of the project requires careful consideration of economic viability during the design phase, while the plans for scaling the operation in the future highlight the need for an easily repeatable design approach. The combination of these factors justifies the implementation of modular construction. On the other hand, this building's function as an educational complex reveals the significance of maintaining indoor environment quality within a comfort range to ensure the students' well-being. Therefore, sunlight utilization, acoustic comfort, and vertical accessibility are the three major factors for design optimization. Besides, urban regulations and emergency evacuation requirements are constraints that must be adhered to.

In order to effectively address the issues mentioned in the problem statement, this study is targeting at finding the answers to the following questions:

1. How does the proposed design methodology address the design optimization goals?
2. To what extent does the proposed generative design methodology demonstrate advancement in terms of design efficiency?
3. Compared with conventional design, how does the proposed methodology succeed in providing economic justification and environmental sustainability?

2. METHODOLOGY

As for methodology, this research employs a variety of software and plug-ins including Rhino 3d, Grasshopper, Pollination (Ladybug Tools), Lunchbox, and Galapagos to create a multi objective weighted genetic algorithm. The mentioned algorithm serves as a flexible and parametric tool for the designers, enabling them to effectively deal with the design challenges and find the optimal point in which the accumulated weighted result of all quantified variables representing design requirements are at the desired maximum. The input data for the proposed algorithm includes the Site boundary curve, annual weather data, Geographical Information System (GIS) information from the Open Street Map (OSM), urban regulation variables, module dimensions and preferred numeric weights for quantified environmental variables which consists of annual sunlight exposure, distance from noise source and accessibility of the indoor space.

Considering urban regulations and module dimensions, a variable landscape of the modular building is constructed which describes the current spatial arrangement of building modules within a three-dimensional environment. Meanwhile, during each iteration of genetic optimization, the selections from the variable landscape are stored in variable arrays, which is the gene pool for the genetic algorithm. Each variable array represents an alternative 3D configuration which would later be analyzed by the grasshopper functions. GIS information from the OSM and Annual weather data are also input data for this analysis. The expected analysis results should reflect the proportion of environmental variables which will later be assigned with numerical weights respectively to form a fitness function whose result is within the range of [0,100].

As for the specific method of multi objective optimization, Simulated Annealing (SA) is applied in this project, which is a highly efficient and resilient method that delivers outstanding solutions for both single and multiple objective optimization challenges. SA could achieve the optimal solution for a single-objective optimization problem and generate a Pareto set of solutions for a Multi objective optimization problem [9]. The concept draws inspiration from the analogy with thermodynamics, specifically the cooling and annealing process observed in metals. Combining genetic algorithms with simulated annealing enhances the probabilistic feedback significantly, leading to reduction of computation time [10]. Although this method is not a new approach for solving complex problems with multiple optimization objectives, applying it under the context of design for modular architecture and construction is considered as the novelty and contribution of this research.

As for the outputs, by using simulated annealing method, the input gene pool would gradually evolve through rapid trial, error and improvement process. The genetic algorithm optimization will provide the final array of numbers representing the selected positions for the modules. The environment variables analysis results for the selected arrays are the optimal results as the goal for the simulation is to achieve 100% for the fitness function within limited time duration. After desired goals are met, design alternatives with the highest optimization score would be compared manually and the most aesthetically appealing option would be selected by the designer based on clients' preference. Finally the clients' preferred option undergoes detailed design and visualization.

3. OPTIMIZATION PROCESS

To address the first research question, which is "clarifying the breakdown structure and quantification methods employed for achieving the optimization goals", this section adopts a simplified example to explain the gene pool of the simulation. Followed by this example, various components of the fitness function could be explored with their proposed structure and effects on the optimization results.

3.1 Variable Landscape

In order to perform Generative Design simulations, it is necessary to provide the algorithm with a set of input variables. These variables are selected building blocks of the fitness function that can be freely modified by the simulation. During each iteration, the simulation will use these variables to adjust the results of the fitness function. The result will then serve as feedback for the adjustments of the next iteration. In the following paragraphs the study will explain the processes of defining the environmental variables, limiting the variable landscape and embedding the urban regulations.

3.1.1 Define the Environmental Variables

In this example, a 60ft. x 60ft flat square will serve as a simplified model of the construction site. Common urban regulations are followed, which vary among different projects based on contexts and functions. But the adjustments of urban regulations could be applied globally. Here are the steps to simulate the generative design algorithm with the necessary inputs:

1. Import the precise boundary curve of the design site into Rhino software.
2. Identify the geographical coordinates of the design site reference points from Global Positioning System (GPS).
3. Import a 3D model of the urban environment with the help of available platforms such as Open Street Map (OSM). Create environmental models using conventional 3D CAD modeling tools.
4. Decide the size of construction modules.
5. Create a reference grid for the placement of modules based on the natural conditions of the construction site.
6. Assign an index for available positions on the reference grid (See Figure 1).
7. Extend the reference grid into the third dimension of the space and assign index for building levels based on design requirements and height limitations.
8. Import the weather data from the closest weather station in Energy Plus Weather (EPW) format provided by "Ladybug Tools". This data will be used for solar radiation analysis and optimization in the later section.

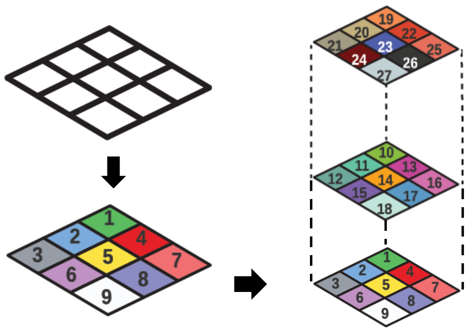


Figure 1: Create the indexed reference grid for the example construction site

3.1.2 Limit the Variable Landscape

After compiling the environment model requirements, we need to define selector arrays (collections of variables) describing the selected grid tiles for arranging the modules in three dimensions. These arrays will form a variable landscape (Gene Pool) for the simulation. This way the algorithm can iterate through various alternatives for the configuration of modules in 3D space and compare the analysis results with each alternative. In this example (see Figure 2), an 5×1 array is created for selecting module positions from a landscape of nine available positions. Considering the exponential impact of repetitive selection and the processing power required for analyzing each alternative, the limitation of variable landscape is necessary.

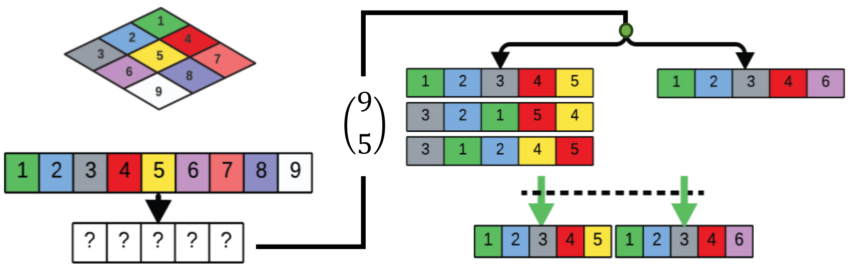


Figure 2: Example selection array with the length of 5 and filtering the repetition

To decrease the calculation time , it's of great necessity to limit the available options to avoid repeated trial/error cycles by the algorithm. Using a selector loop, if a specific tile is selected during a single iteration for placing a module, the mentioned tile will be removed from the variable landscape during the selection stage for that iteration (see Figure 2).

3.1.3 Implement Urban Regulations

To embed the urban regulations into the generative design process, first regulation limits should be defined and broken down into measurable variables. For this example, constraints are as follows (see Table 1).

Table 1. Design constraints

Constraint types	Constraint content	Values
Environmental conditions	Lot dimensions	$60\ ft. \times\ 60\ ft.$
	Lot area	3,600 $sqft.$
Designer decisions	Module dimensions	$20\ ft. \times\ 20\ ft.$
	Total number of positions	27
Urban regulations	Lot coverage of 1st floor	60%
	Occupancy ratio (Total available area for all levels/ Lot area)	220%
	Maximum number of levels	3

To include these constraints in design generation, the algorithm uses a combination of mathematical functions in Grasshopper. These functions use environmental conditions, design decisions and urban regulations as input variables and perform the calculations described below.

$$\begin{aligned} &Available\ area\ of\ 1st\ floor\ = Lot\ area\ \times Lot\ coverage\ of\ 1st\ floor \\ &\qquad = 3{,}600\ sqft.\ \times 60\%\ = \mathbf{2,160\ sqft.} \end{aligned} \tag{1}$$

$$\begin{aligned} &Maximum\ modules\ of\ 1st\ floor\ = Available\ area\ of\ 1st\ floor\ /\ Module\ area \\ &\qquad = \lceil 2{,}160\ sqft.\ /\ (20\ ft.\ \times 20\ ft.) \rceil = \lceil 5.4 \rceil\ = \mathbf{5} \end{aligned} \tag{2}$$

$$\begin{aligned} &Maximum\ modules\ of\ whole\ building \\ &\qquad = Available\ area\ of\ the\ whole\ building\ /\ Module\ area \\ &\qquad = \lfloor 3{,}600\ sqft.\ \times 220\%\ /\ (20\ ft.\ \times 20\ ft.) \rfloor = \lfloor 19.8 \rfloor = \mathbf{20} \end{aligned} \tag{3}$$

Based on these calculations the generative algorithm will be selecting 20 modules to be placed on 27 available positions. 5 modules are on the first level, while 15 modules are on the second and third level combined. These numbers will determine the length of the arrays which will be used for storing various 3D configurations for testing and analysis during each iteration (See Figure 3).

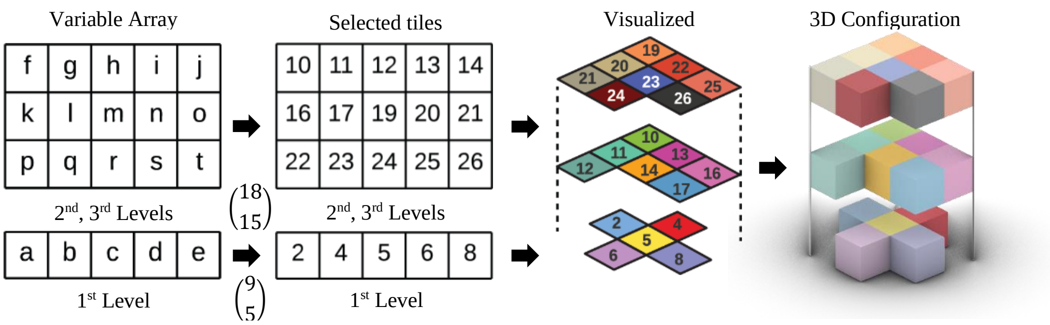


Figure 3: Process of using variable arrays for generating 3D modular configurations

3.2 Fitness Function

As explained in the previous section, a vital component for every generative design optimization is a fitness function. To effectively analyze and compare various configurations suggested by the simulation during each iteration and provide the algorithm with valid and corrective feedback, it is necessary to pay significant attention to various factors of the fitness function, including sunlight exposure, vertical access and integration, and noise pollution. Each factor is broken down into several aspects. The assessment of design for each aspect is through a normalized output value within the range of 0 to 1 with two decimal numbers, which is generated during each iteration of the simulation. If the output value for an aspect is 0.45, it means that that the generated design has the score of 45/100 for the mentioned aspect.

3.2.1 Analyzing Sunlight Exposure

To accurately perform analysis for annual exposure to sunlight for the modules and playground, this study uses a free and open-source repository of functions for Rhino and Grasshopper called "Ladybug Tools." It enables the analysis and visualization and of weather data using Energy Plus Weather (EPW) format in Grasshopper. As displayed in Figure 4, the inputs include the annual weather information from EPW file, specific time in the format of hours in a year (8:00, 10:00, 12:00, 14:00, 16:00 of the first and 15th day in every month), the 3D arrangement of modules and the playground, as well as the context which represents how other modules, neighboring buildings, and objects affect the annual sunlight. Considering the hardware limitations for the simulation it is necessary to limit for time points of daylight. What's more, it is also necessary to consider the effects of other modules during the calculation of annual sunlight during the simulation.

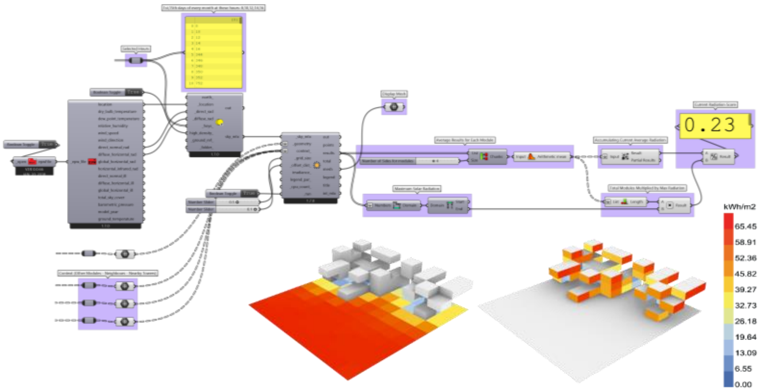


Figure 4: Solar radiation analysis for façade and results snapshot for both aspects

As shown in Figure 4, process of generating the results of **façade** and **playground** is demonstrated. The goal is to maximize the amount of annual sunlight for the building façade and playground with different intentions. The façade should receive the maximum sunlight ultimately in order to guarantee the lighting conditions for the classrooms and living areas. As for the playground, however, as long as the solar energy could satisfy the requirement of sufficient green area construction, the results are acceptable. During each iteration, maximum solar energy for every faces is identified and then multiplied with the total number of evaluated surfaces. This method ensures the final output range from 0 and 1. The highest value of the analysis is 65.45 kWh/m^2.

3.2.2 Embedding Acoustic Comfort

Another significant factor of the fitness function is acoustic comfort. To simplify the analysis of sound interference within limited time, the goal of the optimization is to minimize the importunate noise by maximizing the accumulated distance to adjacent noise sources and the number of obstacles on the path of soundwave transmission. Firstly, to measure the accumulated distance, three streets and an adjacent lot parallel to each side of the construction site are drawn individually. During each iteration, lines are generated between the central point of each module and its closest point to each street line (see the blue, cyan, green, and orange lines in Figure 5). In the next step, the length of these lines is measured, and all of these length are accumulated into a single variable called "accumulated distance for street / lot x". Finally, in order to remap the "Accumulated distance" value for each street or lot, the longest possible distance is determined based on the available module positions. By multiplying this distance with the total number of modules from the urban regulations, the maximum value of the accumulated distance for each street or lot could be calculated. By dividing the accumulated distance to its maximum value, a score withing the range of 0 to 1 is generated. The calculation process is demonstrated in equation (4), in which "m" stands for module number, "n" stands for total number of modules, $d_{x,\,m}$ stands for the distance to segment x (Where segment is representing a street or neighboring lot) for module m, hd_x stands for the distance of the furthest available tile to segment x, AD_x stands for the accumulated distance for segment x, and SD_x sands for the final distance score for segment x.

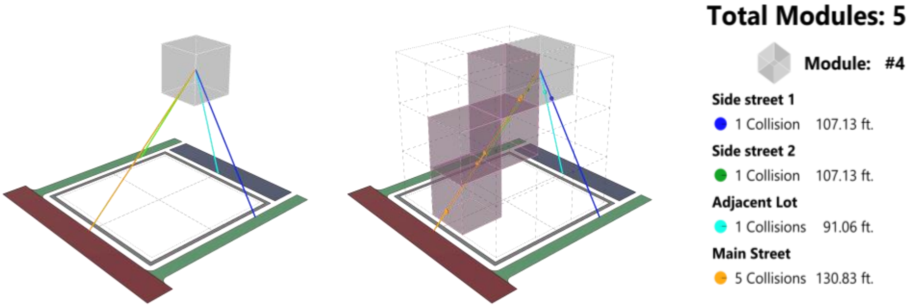


Figure 5: Accumulated distance and collision with obstacles to adjacent noise sources demonstrated in a 5-module configuration

$$SD_x = \frac{(AD_x)}{hd_x} = \frac{(\sum_{m=1}^{n} d_{x,m})}{hd_x} \tag{4}$$

Another proposed method of minimizing noise pollution for each module is through the avoidance of intrusive noise by maximizing the number of obstacles on the path to the noise origin. To achieve this goal, the number of collision points on the colored lines drawn in Figure 5 are recorded and named "Noise path collisions $_{module(m)\,,\, street(x)}$." Then, by accumulating the "Noise path collisions" for all modules related to segment x, "Accumulated collisions for segment x" is calculated.

The number of total available modules for each project is a constant number. Given that each straight line can have a maximum of two collision with a cube, the maximum collisions for every module (the maximum value for Noise path collisions $_{module(i)\,,\, street(j)}$) would be the total number of modules multiplied by two. Consequently if "n" is equal to the total number of modules, the highest value for "Accumulated collisions for segment x" can be determined by using the following formula:

$$Max\ (AC_x)\ =\ 2 \times (n \times (n-1)) \tag{5}$$

In this formula, the path from the central point of each module collides twice with every other modules. By dividing "Accumulated collisions for segment x" with its higher limit, another score is created using number values with two decimals within the range of 0 to 1. The calculation process is demonstrated in equation (6), in which "m" stands for module number, "n" stands for total number of modules, $N_{x,\,m}$ stands for noise path collisions to segment x for module m, AC_x stands for the accumulated collisions for segment x, and SC_x stands for the final collision score for segment x.

$$SC_x = \frac{(AC_x)}{Max\ (AC_x)} = \frac{(\sum_{m=1}^{n} N_{x,m})}{Max\ (AC_x)} \tag{6}$$

3.2.3 Vertical Access and Integration

The third factor of the fitness function is vertical access and integration, contributing to accessibility of this building, especially in emergent situations. In every generative design solution, it is vital for the computer-generated results to satisfy the construction requirements of a real-world building, which is much more than a 3D configuration only exists in the digital world. The significance originates from the fact that feasibility and constructability of the final output must be under consideration. The first aspect evaluated in this part of the fitness function is face connection for the modules. This aspect facilitates the goal of creating an integrated and accessible architectural form. To quantify this aspect, all six faces for every module are evaluated during each iteration in terms of collisions with adjacent modules (see **Figure 6**).

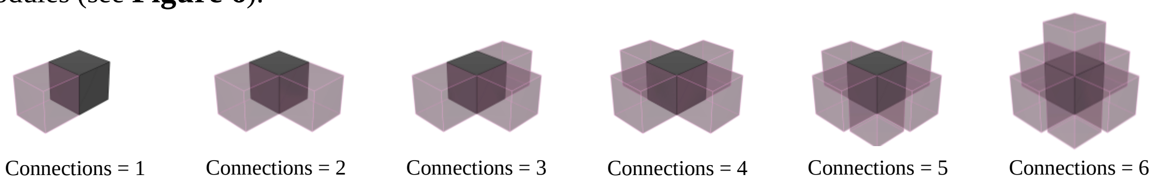


Figure 6: Calculating face connections score

This evaluation produces an integer for each module called "Connections $_{module}$", with a maximum of six and a minimum of zero. To map the result into the range between 0 and 1, a value representing the total number of connections for all modules is created as "Accumulated Connections". By dividing the Accumulated Connections with its maximum value when every module has connections on all six faces, the final score for this aspect is generated. The process is demonstrated in equations (7) where "AC" stands for "Accumulated Connections for all modules", "C" stands for "Connections", "m" for "Module number", "n" for "Total number of modules" and "S_{AC}" is equivalent to final score for face-connection aspect.

$$\mathrm{AC} = \left(\sum_{m=1}^{n} C_m\right), \quad S_{AC} = \left(\frac{AC}{n \times 6}\right) \tag{7}$$

For the second aspect of vertical access and integration, distance between modules is aimed to be minimized. To achieve this goal, distances between modules centers are calculated. For every iteration, the total amount of the center distances is stored in a variable called "Relative Distance $_{module}$". In an approach similar with previous aspects, another cumulative variable named "Accumulated Relative Distance" is then created to store the total Relative Distance for all modules (See Figure 7). To determine the final score within the desired range from 0 to 1, Accumulated Relative Distance is divided by its maximum value. The longest relative distance is between the centers of two corner modules (see Figure 7). Consequently, the maximum value of the relative distance for each module is equal to the product of "longest possible distance" and "total number of modules - 1." The normalization process is similar to the previous sections. The process is demonstrated in equations (8) where "AR" stands for "Accumulated Relative distance", "$R_{h,m}$" stands for "Relative distance", "m" for "Module number", "n" for "Total number of modules", and "S_{AC}" for final module distance score.

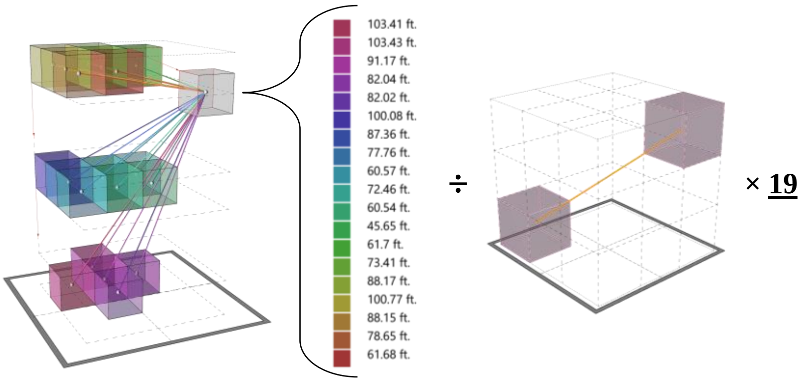
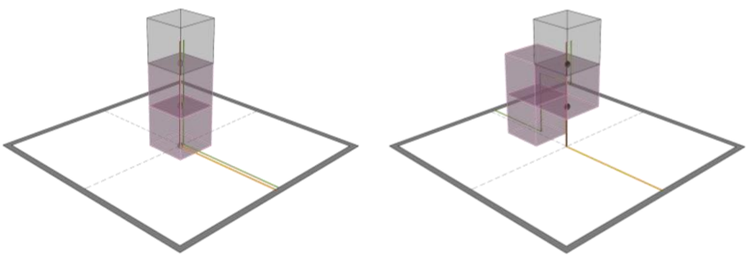


Figure 7: Exploded diagram demonstrating calculation of relative and accumulated distance for an example module (White Cube) in a 20-module configuration.

$$AR = \sum_{h=1}^{n} \sum_{m=1}^{n} R_{h,m}, \quad S_{AR} = \left(\frac{AR}{d \times (n \times (n-1))}\right) \tag{8}$$

In the third aspect of vertical access and integration, the requirements for emergency evacuation are mentioned. Considering the local regulations in terms of maximum capacity for fire and safety evacuation in an educational complex, this building should contain a minimum of four vertical access towers. Additionally, the longest path to the evacuation point shall not exceed 30 meters or 100 feet. In the algorithm, two different methods are used to determine whether these requirements are satisfied. The first method to test vertical access is counting collisions. Based on the height of the module, the prospective level at which the module is going to be placed will be determined. This means that at least for four modules on the third level, the number of collisions for this line (See the orange line in Figure 8) with other modules should be equal to 3. To quantify this process, the number of collisions for the vertical lines connecting every module to the ground with other modules is stored in a variable named "Vertical Access Collisions $_{module}$". Only the modules placed on the 3rd level, the highest level, are analyzed concerning this variable. Throughout the iteration process, if the height of the module center points is bigger than the altitude of level 3 while "Vertical Access Collisions $_{module}$" of that module is equal to 4, the number of Vertical Access Towers (VAT) increase 1. A Boolean variable is employed to test if VAT equals to 4 for normalization.



Config. A 3rd level module
Orange line: 3 Collisions
Green line = Orange line

× Config. B 3rd level module
Orange line: 2 Collisions
Green line ≠ Orange line

Figure 8: Testing vertical access by collision and identifying evacuation path

Finally, the last aspect is minimizing the evacuation path distance for each vertical access tower. This aspect also utilizes the vertical access lines created by the previous aspect as an input. Using the lines with three collisions, the algorithm adds another segment from the touching point of the input lines on the ground to their respective closest point on the side streets (See the green lines in Figure 8). In the next step the algorithm converts the minimization to maximization by calculating the distance for the starting points of each one of these segments to the starting points of other segments respectively, accumulating the results in a variable named "Accumulated Exit Point Distance". To prevent errors caused by calculating these numbers for arrangements with less than four access towers, this process also multiplies its result with a Boolean operator resulting in 0 for any solutions with less or more than 4 access towers. Divide the "Accumulated Evac Point Distance" with its maximum and assign the respective weights, the score of the final aspect of the fitness function could be generated. The process is demonstrated in equations (9) where "AE" stands for "Accumulated Evacuation point distance", "$d(V_{y,x})$" stands for "projected distance on the ground", "x" and "y" for "vertical access tower number", "h" for "Total number of vertical access towers", and "S_{AR}" for final evacuation point distance score.

$$AE = \sum_{y=1}^{h} \sum_{x=1}^{h} d(V_{y,x}), \quad S_{AR} = \left(\frac{AE}{h \times (l + w + o)}\right) \tag{9}$$

3.3 Simulation

The final design includes 45 modules in 4 levels with each module holding 2 levels. After combining different aspects of the fitness function using weighted average method and running Galapagos with simulated annealing for 30 minutes, a score of 66.25% is generated. As demonstrated in Figure 9, the selected design is reflecting the design requirements by maintaining distance with the main street, optimizing the sunlight exposure using voids and creating a unified building with four properly distributed vertical access towers.

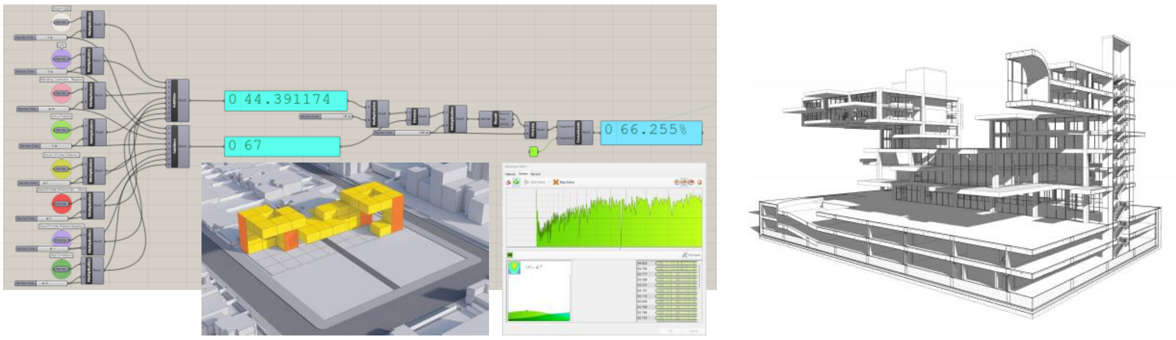


Figure 9: Simulation results in grasshopper - 3d section of detailed design output

3.4 Detailed Design

For visualizing research results and creating architectural documentation, Building Information Modeling (BIM) tools are utilized to transform the digital optimization results into tangible architectural elements (see Figure 10).

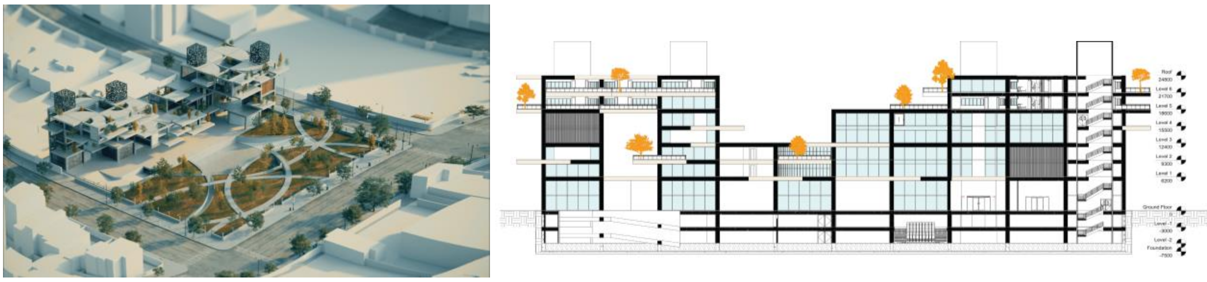


Figure 10: Detailed design of the educational complex

4. CONCLUSION

This research demonstrates a unique approach to generative design for modular construction by creating an optimized educational environment while considering the variable characteristics of different construction projects and facilitating the process of decision making for the designer. The provided methodology in the section 3 can be modified and tailored to the needs of other projects with different capacities. This approach provides quantitative reports, clearly demonstrating the improvements in several aspects of the desired fitness function and design preferences while further optimizing several factors including thermal and acoustic comfort of the design during every iteration of the simulation. In comparison with the conventional approach, if properly utilized, this method can aid designers with a generative optimization tool capable of satisfying various regulations and design requirements of different clients and preferences rapidly and effectively. This method can facilitate design changes and provide the flexibility necessary for modular construction, ultimately providing a solution for the exponential growth of worldwide demands on sustainable and affordable housing for the future. Considering the diversity of requirements for different construction projects, the applications of the simulation can be improved by adding new factors to the fitness function in the future. The capacity of the hardware conducting the simulation is another limiting factor. New and more powerful devices may be capable of conducting more complex calculations with embedded fluid dynamic simulations. This process can also be replicated in other local contexts with different regulations and requirements to improve the generalizability of findings.

5. REFERENCES

[1] R. H. Assaad, I. H. El-Adaway, M. Hastak, and K. L. Needy, "Smart and Emerging Technologies: Shaping the Future of the Industry and Offsite Construction," in *Computing in Civil Engineering 2021*, Reston, VA: American Society of Civil Engineers, May 2022, pp. 787–794. doi: 10.1061/9780784483893.097.

[2] M. Razkenari, Q. Bing, A. Fenner, H. Hakim, A. Costin, and C. J. Kibert, "Industrialized Construction: Emerging Methods and Technologies," *Comput. Civ. Eng. 2019 Data, Sensing, Anal. - Sel. Pap. from ASCE Int. Conf. Comput. Civ. Eng. 2019*, vol. Volume, no. Number, pp. 352–359, 2019, doi: 10.1061/9780784482438.045.

[3] C. Turner, J. Oyekan, and L. K. Stergioulas, "Distributed manufacturing: A new digital framework for sustainable modular construction," *Sustain.*, vol. 13, no. 3, pp. 1–16, 2021, doi: 10.3390/su13031515.

[4] S. Jung and J. Yu, "Design for Manufacturing and Assembly (DfMA) Checklists for Off-Site Construction (OSC) Projects," *Sustain.*, vol. 14, no. 19, 2022, doi: 10.3390/su141911988.

[5] A. Darko, A. P. C. Chan, M. A. Adabre, D. J. Edwards, M. R. Hosseini, and E. E. Ameyaw, "Artificial intelligence in the AEC industry: Scientometric analysis and visualization of research activities," *Autom. Constr.*, vol. 112, p. 103081, Apr. 2020, doi: 10.1016/j.autcon.2020.103081.

[6] S. BuHamdan, A. Alwisy, and A. Bouferguene, "Generative systems in the architecture, engineering and construction industry: A systematic review and analysis," *Int. J. Archit. Comput.*, vol. 19, no. 3, pp. 226–249, Sep. 2021, doi: 10.1177/1478077120934126.

[7] M. Wasim, P. Vaz Serra, and T. D. Ngo, "Design for manufacturing and assembly for sustainable, quick and cost-effective prefabricated construction – a review," *Int. J. Constr. Manag.*, vol. 22, no. 15, pp. 3014–3022, Nov. 2022, doi: 10.1080/15623599.2020.1837720.

[8] A. Lambora, K. Gupta, and K. Chopra, "Genetic Algorithm- A Literature Review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, IEEE, Feb. 2019, pp. 380–384. doi: 10.1109/COMITCon.2019.8862255.

[9] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *J. Oper. Res. Soc.*, vol. 57, no. 10, pp. 1143–1160, Oct. 2006, doi: 10.1057/palgrave.jors.2602068.

[10] I. Jeong and J. Lee, "Adaptive simulated annealing genetic algorithm for system identification," *Eng. Appl. Artif. Intell.*, vol. 9, no. 5, pp. 523–532, Oct. 1996, doi: 10.1016/0952-1976(96)00049-8.