Flexible Plot-scale Urban Design using Quadratic Programming

Qian Hu[1], Yujiao Wang[2], Peng Tang[3]

Abstract. This research aims to tackle the problem of the generative urban design of residential areas using a general solving machine of mathematical programming. Residential areas on Chinese university campuses are taken as examples. As a type of urban design problem, the layout of residential areas on campuses is subject to multiple indicators and various boundary shapes. Quadratic Programming (QP) offers an representation to this problem, and with the assistance of cutting-edge mathematical programming solvers, the urban design problem with quadratic constraints can be automatically tackled. However, the difficulties lie in formulating complex boundaries, flexible building templates, and directional variability. To overcome these challenges, this research combines inside-model techniques of representation and outside-model modules utilizing geometric methods to enhance the main model of QP. The QP model incorporates multiple building rules and complex boundaries, while extended methods of the oriented bounding box and vector field method are employed to address directional problems. A pipeline is provided to apply the approach in real urban design projects. The generated results validate the effectiveness of the enhanced model and the pipeline.

Keywords: Generative Urban Design; Universal Solving Machine; Mathematical Programming; Quadratic Programming; Residential Areas on Campus; Geometric Methods;

---

[1] Qian Hu
e-mail: huqian@seu.edu.cn,

[2] Yujiao Wang
e-mail: yujiaowang1230@gmail.com

[3] Peng Tang (✉)
School of Architecture, Southeast University,Nanjing210096, China. Key Laboratory of Urban and Architecture Heritage Conservation (Southeast University), Ministry of Education, Nanjing, 210096 .
e-mail: tangpeng@seu.edu.cn,

1 Introduction

Generative urban design refers to the use of generative methods in urban design scenarios and has attracted significant attention from academic communities, professionals, and government agencies in recent years (Jiang et al., 2023). Generative urban design can benefit from the recent development of universal solvers, which can automatically tackle linear and other related mathematical programming problems (Meindl & Templ, 2012). The approach has been approved to be useful in dealing with design problems(Kamol Keatruangkamala and Krung Sinapiromsaran, 2005). This frees designers from having to Fig. out how to solve problems and allows them to focus on what to solve. In other words, designers are only required to formulate urban design problems with mathematical expressions, without calculating. However, the cost is that the problem description must be highly formalized under specific rules of the universal solver for efficient problem-solving.

Quadratic Programming (QP) is a type of mathematical programming with quadratic constraints that can be solved using a cutting-edge universal solver (Gurobi Optimization Inc., 2012). Compared to other methods such as evolutionary algorithms, QP shows better performance in adhering to explicit constraints. It has been applied to urban design scenarios with a data structure of special ordered set (Hua et al., 2019), where constraints of irregular boundaries are bypassed, and fixed templates are used to represent buildings. Other generative design using mathematical programming focused on floor plans and thus did not have to face complex shapes(Wu et al., 2018). A break-though regarding complex boundaries has been reached with highly modularized data-structure, but it is consequently not flexible for real-world size(Peng, Yang and Wonka, 2014). To extend the flexibility of using QP in generative design may be the next challenge to tackle.

Designing residential areas on Chinese university campuses, including placing buildings, is a typical example of urban design with multiple constraints, such as boundaries, floor area ratios, building density, distances between buildings etc. Generative design methods can improve the effectiveness and efficiency of this design process, and a QP solver can be used for decisions under urban planning indicators not directly related to operational knowledge of urban design. However, it is challenging to deal with complex residential area shapes and the flexibility of dormitory buildings with a QP model and to generate results similar to manual design schemes.

This research aims to enhance the flexibility when using QP to solve design problems, through a solution for generative design of residential areas on Chinese university campuses. By formulating main considerations inside the QP model and providing extra modules outside, the problem is decomposed and reorganized, generating valuable results for real design practice. The programs are written in

Python with Gurobi used as the mathematical programming solver, and the results are visualized using Rhino and AutoCAD.

2 The main model

Designing residential areas on Chinese university campuses poses integrated problems where the boundaries' shapes, floor area ratios, building densities, and distances between buildings impose strong restrictions on the design solutions. In the meantime, the locations, heights, and lengths of dormitory buildings can be flexible within permitted domains. According to experienced designers, the manual design is an iterative process, where the solution moves from one state to another, until feasible solutions are found. To supersede this process, Quadratic Programming (QP) is introduced.

QP is a type of mathematical programming that formulates all constraints with equations and inequalities with degrees not exceeding 2. With the aid of cutting-edge solvers, problems with quadratic constraints can be tackled automatically. While the requirement makes problem formulation more challenging, it also promote the efficiency of solution-finding.

2.1 Building formulation

In the residential areas, the majority of buildings are dormitories (Fig.1(a)) characterized by a corridor-dominated structure. The depth of a building is determined by the presence of rooms on both sides of the corridors, while the length is dependent on the number of room units in each row. The building height is controlled by the number of above-ground floors, and the distance between buildings in the main direction is closely related to the height to ensure adequate natural lighting and direct sunlight for most of the rooms.

In addition to the strip-shaped buildings with a corridor-dominated structure, there are also box-shaped buildings with relatively stable sizes, such as the canteen building (Fig. 1(b)). These buildings typically have no more than three floors, with similar depths and lengths.

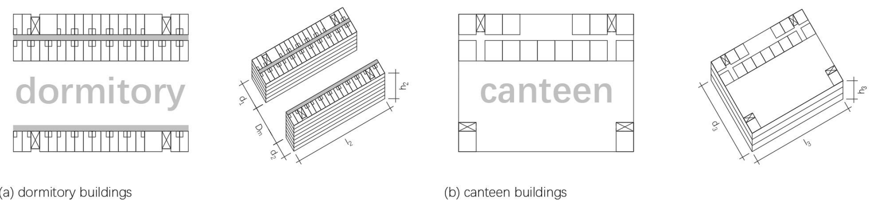

Fig. 1 Dormitory building and canteen building

To define a dormitory building, we need variables (x, y) for its location and (d, l, h) for its size. To further represent the variability and fixation of the building, (d, l, h) are replaced through the below formulas.

$$d = d_{cor} + (1+\xi)\times d_{room} \qquad (\xi \in \{0,1\}) \tag{2.1}$$

$$l = n_l \times l_{unit} \qquad (n_h \in N^*) \tag{2.2}$$

$$h = n_h \times h_{unit} \qquad (n_h \in N^*) \tag{2.3}$$

Here n_h and n_l are the main variables. At the meantime, the range of the main variable is limited according to real design needs. For instance, $5 \le n_h \le 6$ and $20 \le n_l \le 27$.

For the sake of convenience, the definition of a canteen building is extended from a dormitory building, while some constants should be changed and the variables n_h and n_l are usually fixed or fluctuating in narrow ranges.

2.2 Building relationships

Once the buildings are defined, the primary task is to describe the relationships between them, with the main one being to maintain distance. Considerations include fireproofing distance and sunlight distance based on Chinese regulations. The former requires the distance in any direction (D_x) should be more than 6, 9 or 13 m, according to specific features of the buildings. The latter requires the distance between buildings in the main direction (D_m), usually south-north, to be more than the minimum distance for adequate direct sunlight for the building in the north. The formula is below.

$$D_m \ge h \times k \tag{2.4}$$

The value of constant k is determined by the location of the city, and it is mainly 1.35 in this study. However, if the orientation of the buildings does not face south, local regulations may give slight variations in this value.

To express these requirements in a QP model, we introduce the concept of a control area (Fig. 2) which utilizes variables $(D_{iN}, D_{iS}, D_{iW}, D_{iE})$ to expand a building's shape to include its actual area of influence. The central principle is that each building must not encroach on another building's control area, while allowing for overlap between control areas. For example, when considering building-1 and building-2:

$$\sum_{i=1}^{4} \xi_i = 1 \qquad (\xi_i \in \{0,1\}) \tag{2.5}$$

$$\begin{aligned} &\xi_1 \times (x_i - x_j - l_j - D_{jE}) + \xi_2 \times (-x_i + x_j - l_j - D_{jW}) + \\ &\xi_3 \times (y_i - y_j - d_j - D_{jN}) + \xi_4 \times (-y_i + y_j - d_j - D_{jS}) \geq 0 \end{aligned} \tag{2.6}$$

Here, binary variables ξ_i are used to guarantee building-i would keep away from the control area of building-j from either N/S/W/E side. Also note that building subscripts (i and j) are exchangeable, which means two groups of constraints are set for every building pair.

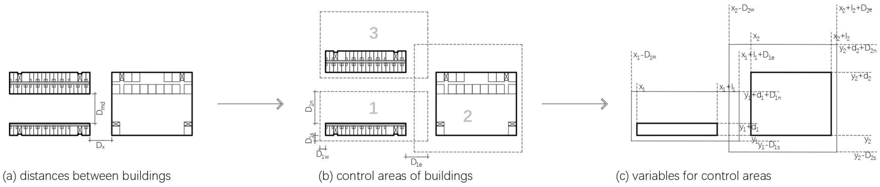

Fig. 2 Control areas of buildings

2.3 Boundary formulation

The aim of this section is to illustrate how QP models can represent various shapes of boundaries (polygons) and their relationships with building rectangles. Different boundary shapes for residential areas are demonstrated in Fig. 3, and we do not consider holes in polygons in this section.

To confine a building to a specific area, we can approximately constrain its key sample points, which include the four corner points. In relatively complex situations, we may need to extend these points to 12 or more to avoid errors (Fig. 3(d)). This is achieved by formulating inequalities for each sample point $P(x, y)$ and a polygon $\{Q_1, Q_2, Q_3, ..., Q_n\}$. Note that these expressions are valid through this section.

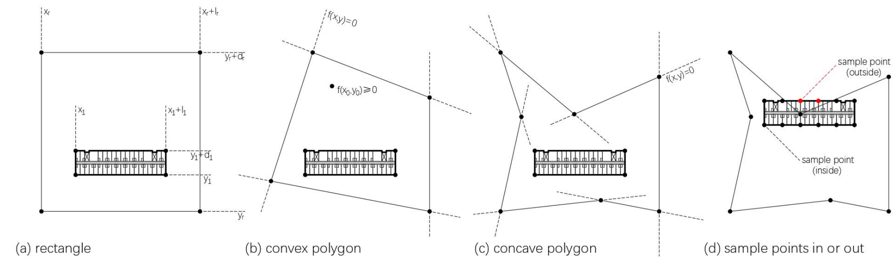

Fig. 3 Different shapes of residential areas

The simplest shape to consider is a rectangle, and the corresponding inequalities are easy to write. More generally, for an irregular convex and a sample point, the formula is 2.7 (representing the equation of line Q_iQ_{i+1} by $a_ix + b_iy + c_i = 0$). Take the opposite numbers for (a_i, b_i, c_i) in the expression of Q_iQ_{i+1} when $a_ix_0 + b_iy_0 + c_i \geq 0$ does not work for the center point $Q_0(x_0, y_0)$ of the convex polygon. The inequalities for not-in-convex, namely to represent that a sample point $P(x, y)$ is not in a convex polygon $\{Q_1, Q_2, Q_3, ..., Q_n\}$ is 2.8. The rules for taking opposite numbers are the same as in-convex.

$$\forall i \in \{1,2,3,...,n\}, a_ix + b_iy + c_i \geq 0 \tag{2.7}$$

$$\sum_{i=1}^{n} \xi_i = 1 \quad (\xi_i \in \{0,1\}) \qquad \wedge \qquad \sum_{i=1}^{n} \xi_i (a_ix + b_iy + c_i) \leq 0 \tag{2.8}$$

The problem is more complex for concave polygons. The equations would likely exceed the limitation of polynomial degrees (no more than 2). To overcome this difficulty, we use a concavity processing method. As shown in the below codes and Fig. 4 , a concave polygon is transformed into IN convex polygon and NOT-IN convex polygons:

WHILE $\exists Q_i \in C_0, < \overrightarrow{Q_{i-1}Q_i}, \overrightarrow{Q_iQ_{i+1}} >< 0$

  IF $\exists a,b \in \{1,2,3,...,n\}, a \leq b, \forall i \in \{a, a+1, ..., b\}, < \overrightarrow{Q_{i-1}Q_i}, \overrightarrow{Q_iQ_{i+1}} >< 0$

  $\wedge < \overrightarrow{Q_{a-1}Q_a}, \overrightarrow{Q_aQ_{a+1}} > \geq 0 \wedge < \overrightarrow{Q_{b-1}Q_b}, \overrightarrow{Q_bQ_{b+1}} > \geq 0$

    $C_0 = C_0 - \{Q_a, Q_{a+1}, ..., Q_b\}, T = T \bigcup \{\{Q_{b+1}, Q_b, ..., Q_{a-1}\}\}$

  END

END

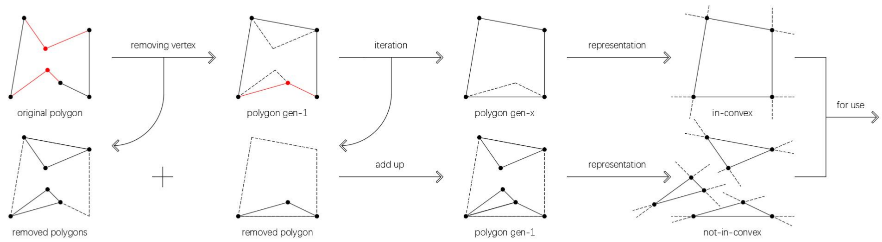

Fig. 4 Concavity Processing Method

This process is iterated until the polygon becomes convex. The resulting in convex polygon and NOT-IN convex polygons can be represented in a mathematical program respectively, allowing for accurate descriptions that do not exceed the degree limitation.

2.4 Overall constraints and solving

The cutting-edge solver has the capability to provide optimal solutions, such as maximizing the floor area, if an objective is defined. Alternatively, it can generate possible solutions if the objective is set to be nonsense. In section 2.1, we have already considered the variables and constraints related to individual buildings and the relationships between buildings. However, there are still some overall indicators that need to be inputted before the solver is executed.

These overall indicators include the floor area ratio (F), building density (δ), and height limitation. The consideration for overall height is the same as the height requirements for individual buildings. For {builing-1, builing-2, ⋯ , building-n}, the other two can be formulated as below.

$$\delta_{\min} \leq \sum_{i=1}^{n} d_i \times l_i \leq \delta_{\max} \tag{2.9}$$

$$F_{\min} \leq \sum_{i=1}^{n} d_i \times l_i \times n_{hi} \leq F_{\max} \tag{2.10}$$

As shown in Fig. 5,We have generated some schemes, using real-world conditions in Nanjing, China. These buildings are designed to have rows of rooms on both sides of the corridor, which can be called double-depth.

Also, the results and run-times under different conditions have been tested, as is shown in Fig. 6.

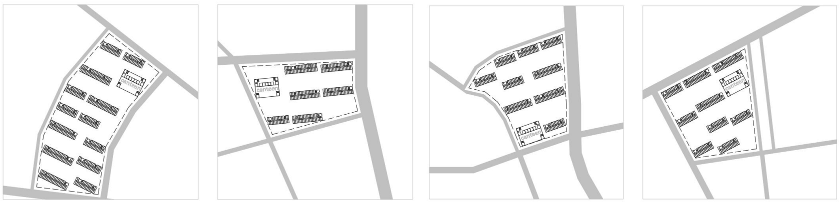

Fig. 5 Possible solutions for residential areas in Nanjing city

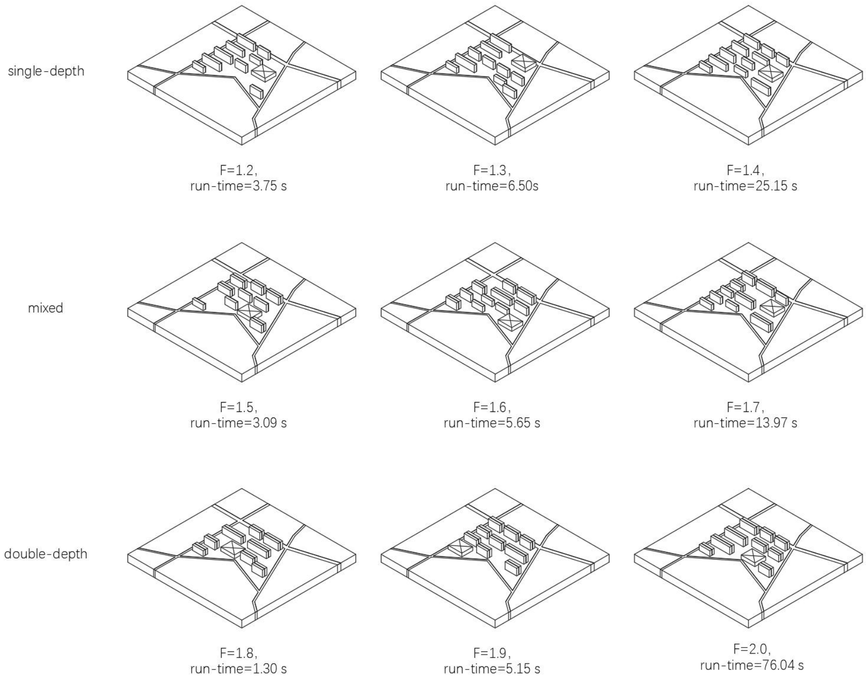

Fig. 6 The results and run-time of solving under different circumstances

3 Extra modules

3.1 Building alignment

Buildings tends to be grouped or aligned in urban design, which can be easy to describe in a grammar-based generative system. Here, QP is proved to be similarly helpful in keeping specific types of alignment.

In Fig. 5 and Fig. 6, it can be observed that the patterns of these buildings are relatively casual. This is because they do not adhere to any specific grammar, but instead respond reasonably to the complexity of the boundaries.

We conducted an additional experiment to illustrate that the QP solver could also control building alignment in a similar way to rule-based approaches. As is shown in Fig. 7, a specific alignment rule is extracted from an example and applied to another using QP. Fig. 7(a)(b) shows a typical example of how building alignment works in an irregular area. The buildings are grouped in pairs and aligned on at least one side, allowing the other side to respond freely to the irregular boundaries. The equations and inequalities for building-i and building-j are given below, which are grouped in pairs and align to each other.

$$\xi_a + \xi_b = 1 \qquad (\xi_a, \xi_b \in \{0,1\}) \tag{3.1}$$

$$\xi_a \times (x_i - x_j) + \xi_b \times (x_i - x_j + l_i - l_j) = 0 \tag{3.2}$$

$$y_i - y_j \le 0 \tag{3.3}$$

$$2 \times k \times h_{unit} \times (n_h)_{\min} + y_i - y_j \ge 0 \tag{3.4}$$

The result generated for another area can be seen in Fig. 7(c)(d), where the buildings are designed to have a row of rooms on the south side of the corridor, which can be called single-depth. Fig. 8 compares the result without and with alignment requirements. It can be seen that the grouping method works both for single-depth buildings and double-depth buildings.

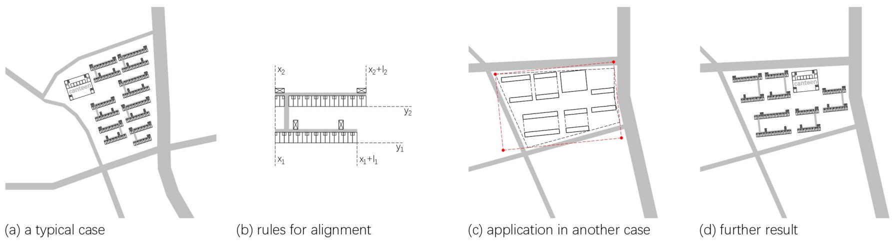

Fig. 7 The process of extracting and applying alignment rules from a case to another

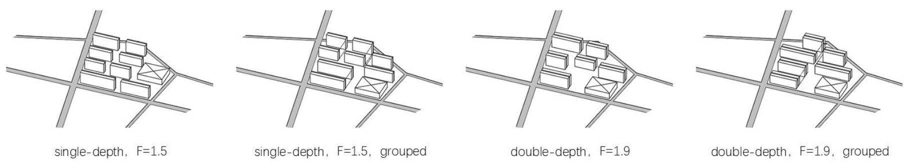

Fig. 8 Comparison between normal results and 'grouped' results

3.2 Building redirection

This section provides and extra module to redirect the buildings. Though this is not direct coordination to the QP model but a subsequent step, it prove that QP provide an excellent initial stage for other generative design methods.

In the previous sections, we utilized the QP model to effectively generate potential solutions, and made great efforts to extend its capabilities. One of the key points was to reduce the degrees of the expressions to no more than 2. While this approach may work efficiently for polynomial expressions and some of the logical transformations, it is not suitable for problems regarding directions, which would inevitably require approximate representation and consume a significant amount of time.

As depicted in Fig. 9, designers use various methods to assign directions to buildings in residential areas. It is apparent that QP can hardly work for a multi-directional case. However, this can be solved with an extra step.

Fig. 9 Directional Methods of residential area Design

When residential areas exist to be multi-directional, design decisions for each building's direction are independent of others' and related to its positions. Therefore, a continuous and smooth mapping $P(x, y) \rightarrow \theta \in [0, \pi / 2)$ is required for all the directions of the buildings, ensuring their cohesion with the boundary.

A method of vector field is thereby introduced for this purpose, which has been applied in street modelling (Chen et al., 2008), building mass generation (Sun & Dogan, 2022), and been extended to natural ventilation in urban design (Zhang et al., 2023).

There are diverse ways to formulate smooth mapping, and this research utilizes a method which is the same as Zhang's. Fig. 10 shows this process and the obtained mapping, namely the vector field is then used to slightly coordinate the preliminary results generated in section 2.4, leading to the results shown in Fig. 11. Here the distances between buildings might be slightly changed and the sunlight may be influenced for some buildings. The corresponding solutions might be pre-defining bigger distances or arranging self-organizing system for the buildings. As this section is mainly used to roughly illustrate the possibility of subsequent process of QP, more enhancement may not be discussed here.

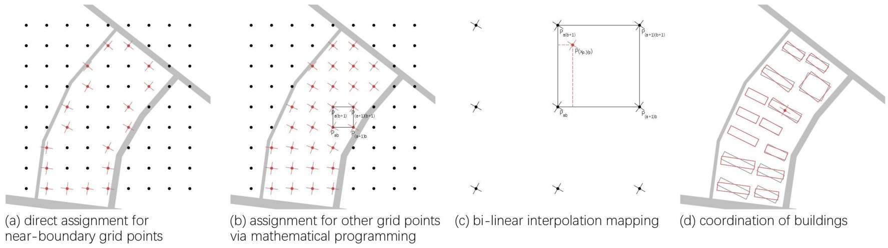

Fig. 10 Steps for multi-direction mapping

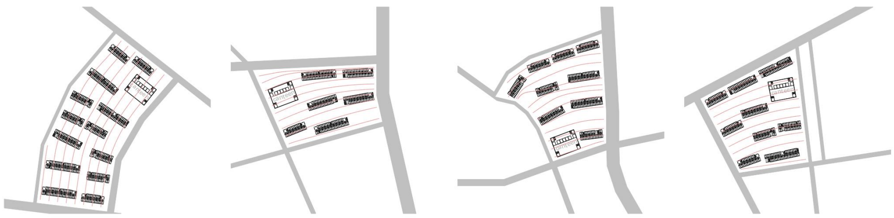

Fig. 11 Results after modification in vector field

3.3 Pipeline organization

Additionally, we provide a pipeline to illustrate how these modules can be combined to produce schemes. Fig. 12 presents an overview of a modeling pipeline, which demonstrates a possible way to integrate the QP model and geometric methods discussed in sections 3.1 to 3.3, for real design scenarios. The pipeline allows users to automatically generate buildings by defining key parameters and area boundaries. Constraints are formulated in a QP model and then solved by a mathematical programming solver. Geometric methods are applied both before and after the main model to enhance the framework's flexibility and applicability.

This pipeline has been successfully tested in several campus design projects, enabling efficient building generation in residential areas based on preliminary land layout schemes. Information about ongoing campus design projects is not disclosed in this paper to preserve confidentiality.

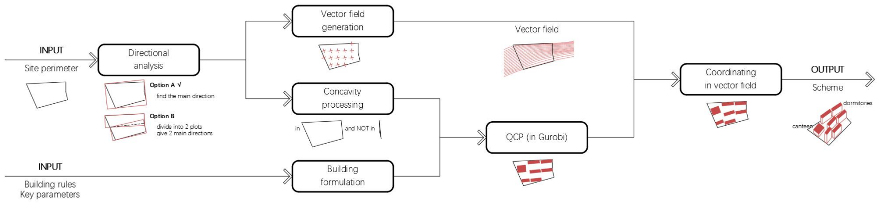

Fig. 12 A Modelling Pipeline

4 Conclusion

This paper demonstrates the effectiveness of utilizing QP and cutting-edge solvers to address the urban design issues prevalent in residential areas within Chinese universities. By incorporating inside-model techniques and outside-model modules that employ geometric methods, the model can effectively handle complex boundaries, flexible building templates, and directional variability. The flexibility of QP models and general solvers is apparently enhanced. This approach, which is founded on a comprehensive understanding of urban design problems, has proven to be beneficial in practical urban design scenarios.

However, it is worth noting that this study does not fully capture the flexibility of buildings that are primarily dominated by corridors, as rule-based approaches can do. Moreover, the program's speed may be compromised when dealing with large-scale problems.

References

Chen, G., Esch, G., Wonka, P., Müller, P. and Zhang, E. (2008). Interactive procedural street modeling. ACM Transactions on Graphics, 27(3), p.1. doi:https://doi.org/10.1145/1360612.1360702.

Guo, Z. (2021). From simulation to systhesis: Architectural modeling with context-based encoding using data-driven computational machines. [D] Available at: https://doi.org/10.3929/ethz-b-000527259.

Hua, H. and Dillenburger, B. (2023). Packing problems on generalised regular grid: Levels of abstraction using integer linear programming. Graphical Models, [online] 130, p.101205. doi:https://doi.org/10.1016/j.gmod.2023.101205.

Hua, H., Hovestadt, L., Tang, P. and Li, B. (2019). Integer programming for urban design. European Journal of Operational Research, 274(3), pp.1125 – 1137. doi:https://doi.org/10.1016/j.ejor.2018.10.055.

Jiang, F., Ma, J., Webster, C.J., Chiaradia, A.J.F., Zhou, Y., Zhao, Z. and Zhang, X. (2023). Generative urban design: A systematic review on problem formulation, design generation, and decision-making. Progress in Planning, [online] p.100795. doi:https://doi.org/10.1016/j.progress.2023.100795.

Kamol Keatruangkamala and Krung Sinapiromsaran (2005). Optimizing Architectural Layout Design via Mixed Integer Programming. In: Computer Aided Architectural Design Futures 2005. pp.175 – 184. doi:https://doi.org/10.1007/1-4020-3698-1_16.

Meindl, B. and Templ, M. (2012). Analysis of commercial and free and open source solvers for linear optimization problems. repositum.tuwien.at. [online] Available at: http://hdl.handle.net/20.500.12708/37465 [Accessed 18 Dec. 2023].

Peng, C.-H., Yang, Y. and Wonka, P. (2014). Computing layouts with deformable templates. ACM Transactions on Graphics, 33(4), pp.1 – 11. doi:https://doi.org/10.1145/2601097.2601164.

Sun, Y. and Dogan, T. (2022). Generative methods for Urban design and rapid solution space exploration. Environment and Planning B: Urban Analytics and City Science, p.239980832211421. doi:https://doi.org/10.1177/23998083221142191.

Wang, X., Song, Y. and Tang, P. (2020). Generative urban design using shape grammar and block morphological analysis. Frontiers of Architectural Research. doi:https://doi.org/10.1016/j.foar.2020.09.001.

Wu, W., Fan, L., Liu, L. and Wonka, P. (2018). MIQP-based Layout Design for Building Interiors. Computer Graphics Forum, 37(2), pp.511 – 521. doi:https://doi.org/10.1111/cgf.13380.

Zhang, Q., Li, B., Li, H. and Tang, P. (2023). Towards Integration and Hybridization in Urban Generation: An extendable urban generative system for better natural ventilation. doi:https://doi.org/10.52842/conf.ecaade.2023.2.379.