URBAN SITE ADAPTIVE LAYOUT GENERATION:USER-CONFIGURABLE ALGORITHMS BASED ON MQP

YUJIAO WANG[1], QIAN HU[2] PENG TANG[3*], RONG FANG[4] and YUAN MENG[5]
[1,2,3]*School of Architecture, Southeast University, Nanjing 210096, China.*
[3] *Key Laboratory of Urban and Architecture Heritage Conservation (Southeast University), Ministry of Education, Nanjing, 210096 .*
[4,5] *Southeast University Architectural Design and Research Institute Co., Ltd, Ministry of Education, Nanjing, 210096 .*

[1]*yujiaowang1230@gmail.com, 0009-0009-3836-1219*
[2]*huqian@seu.edu.cn, 0009-0007-0532-3473*
[3]*tangpeng@seu.edu.cn, 0009-0003-1658-6774*
[4]*4277190@qq.com, 0009-0000-4345-7121*
[5]*522786608@qq.com, 0009-0002-1756-5676*

Abstract. The growing reliance on digital methods presents a promising avenue for addressing challenges related to maximizing design quality according to user preference. This paper introduces a suite of integrated algorithms based on Mixed Quadratic Programming (MQP) to enhance the adaptability of the existing model for generative layout challenges in urban design. The algorithms for site handling are divided into three key directions: (1) Boundary Adaptive, (2) Interaction with Existing Elements, (3) Flexible Road Network. These directions are implemented through a combination of hard constraints and soft constraints. The resulting layout model, incorporating these algorithms, is user-configurable to meet specific project requirements. To validate the effectiveness of this approach, we applied the algorithms in a campus planning project. The results demonstrate remarkable efficiency and precision in generating layouts that include plots and design elements. These findings affirm the robust applicability of our algorithms for creating urban-scale site layouts.

Keywords. Generative urban design, Quadratic Programming, layout modelling, site adaptation, user-configurable

1. Introduction

Urban designers face complex decision-making processes, considering design requirements, construction indices, and contextual factors while striving to optimize the urban environment. The complexity of building layouts can be addressed through computer algorithms that extract rules and goals, allowing for disassembly, de-

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 2, 505-514. © 2024 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

abstraction, and partial automation (Jiang et al., 2023). The architectural layout problem, a spatial configuration challenge, involves determining the positions and sizes of related objects to satisfy requirements and maximize design quality (Galle, 1981). At different scales, they both similarities and peculiarities. Most plan layout problem can be viewed as parcel layout according within the scale hierarchy of urban form elements (Kropf, 2013). Among them, urban layout considers more about the context including transportation, function and form structure compared with small scale layout like floorplan generation.

In mid-scale layouts, where the challenge lies in arranging plots with various functions within a fixed urban site while optimizing transportation, existing research has employed various algorithms. Categorizing these algorithms based on cell construction methods reveals different approaches: (1) Splitting plots to fill the domain without gaps, achieving a hierarchical layout through road and plot splitting (Li et al., 2015; Y.-L. Yang et al., 2013). This approach can be adapted to different site environments, but lacks precision in constraining the plot metrics and the topological relationships between them. (2) Template fill involves occupying discrete grids with pre-defined templates, adaptable to the non-orthogonal system (Peng et al., 2014). It requires abstracting parcels into a limited number of type templates for reuse. (3) Resizing box strictly constrains the size, position, and adjacency of parcels, considering optimization objective (Bao et al., 2013). This approach is suitable for interactive urban layouts with explicit requirements, but still limited in orthogonal domain. (4) Case-based approached, leveraging machine learning, reduce human labour in rule transformation (Rhee & Veloso, 2021). However, the black box nature presents challenges in user interaction and precise control.

In summary, existing research often extract abstract problems such as form simulation, template-based arrangements, or layout in simplified orthogonal sites, limiting its applicability in complex urban land use allocation. To accommodate real-world site conditions, this study explores site layout generation at the urban design scale, introducing a user-configurable approach within the framework of the Quadratic Assignment Problem (QAP).

2. Framework

This paper introduces a refined framework for site handling that addresses several key limitations and expectations, which can be divided into three main directions:

Boundary Adaptive: Common site shapes, such as complex rectangles, non-orthogonal polygons, and irregular configurations, are categorized. Constraints are then formulated to ensure that parcels fit within the boundary while preventing overlap with existing roads and landscapes.

Interaction with Existing Elements: Existing elements, including entrances, centre axis and road, are extracted and represented by their coordinates. Their typological relationships with parcels are summarized as near, away, on the one side, inside or outside. Algorithms are developed to achieve these relationships.

Flexible Road Network: Road structures, such as grid-type, branch-type, and ring roads, are treated as variables. Their dynamic relationship with parcels is translated to synchronize their generation, accommodating different road structures in site layouts.

To achieve them, a set of integrated algorithms are provided based on a combination of hard constraints and soft constraints according to the extent of mandatory. The user-configurability of these algorithms is a key feature, allowing for customization and flexibility in addressing the complexities of diverse urban design projects.

2.1. THE BASIC QAP ALGORITHMS

One of the earliest applications of QAP in architectural layout formulated optimal layout design as a multi-objective mixed-integer programming model (Keatruangkamala & Sinapiromsaran, 2005). Rooms or plots were represented by rectangular cells, where each cell *Di* was parameterized by position and size attributes as: $D_i = [x_i, y_i, w_i, h_i]$. For this study, continuous variables are employed and basic dimension constraints are captured using linear and quadratic formulas, including the range of height, width, coordinates, aspect ratio, and area of each cell within a layout Γ (see Figure 1(a)). On the basis of non-overlap constraint (Medjdoub & Yannou, 2000), a minimal distance of *t* is maintained to preserve space for roads during further generation in this paper (see Figure 1(b)). Besides, connectivity constraints are formulated oppositely to assure alignment of cell *j* with cell *i* on one of the four sides.

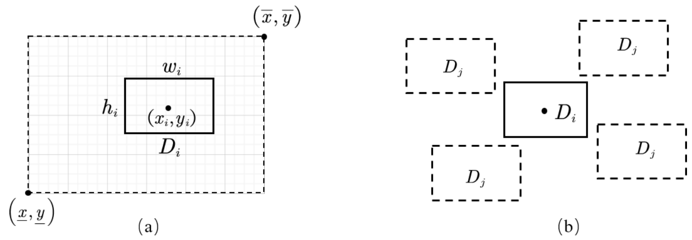


Figure 1. Construction of cells: variables and primary constraints

2.2. CONSTRAINTS TYPE

Besides, there are other constrains according to specific sites and requirements, with various extends and may be contradictory with each other. ‘Hard constraints’ and ‘soft constraints’ are used in optimisation to express two types of constraints (Bao et al., 2013), among which the application of soft constraints provides a feasible way for the designer to convert the design intent into the design optimization process. For this paper, the requirements on the dimension and position are separated as the two types of constraints according to their semantics (see Table1). Hard constrains are for compulsory limitation (i.e., cells are limited in the site domain) which are achieved by equation or inequation. Soft constraints working for subjective design preference and flexible targets in layout can be abstracted as energy functions $E(\Gamma)$ for the layout Γ, with lower energy indicating more desirable layouts. The energy function consists of soft constraints of each parcel (i.e., maximize the area of certain parcel) and overall constraints (i.e., Minimize the unused land), which are expressed as $E(D_i)$ and $E_v(\Gamma)$ respectively with user-configurable weight. The model based on MQP is expressed as:

$$\text{s.t. } G(x) \leq g$$

$$\min E(\Gamma) = \sum_{i=0}^{n} E(D_i) + \sum_{v \in V} E_v(\Gamma)$$

Table 1. Description of hard constrains and soft constrains

Hard constraints	**Soft constraints**
Area domain	Minimize the unused land
Size domain	i close to Road
Ratio of length and width	i far from j
Layout domain	i close to Enter
Distance between cells	Maximize the area of i
i align Road	Minimize the area of i
i align to j	…

3. Site Adaptive Algorithms

3.1. BOUNDARY ADAPTIVE ALGORITHMS

Common urban sites are categorized into three types and corresponding algorithms are proposed for establishing hard constraints separately.

3.1.1. Complex orthogonal boundary

An orthogonal polygonal layout domain can be described by a set of rectangles, including the bounding box subtracted by rectangles *obstacle*, which cannot be covered by any parcel (Wu et al., 2018). Layout is performed with the constraint that parcels are within the bounding box A and outside the subtracted rectangles (by the non-overlapping constraints described earlier). This algorithm also shows high efficiency in the tests of this paper.

3.1.2. Non-orthogonal boundary

Some of the site boundaries in cities conform to the surrounding roads forming e.g. trapezoids, irregular quadrilaterals and other polygons, where the non-orthogonal edges can be expressed as linear analytical formulas constraining the parcels. The exact method varies depending on the concavities and convexities of the boundary polygon being formed.

In a convex polygon site, constrains are within the AABB and set to keep all corners of each parcel to be on one side of the line $L = [x, y | ax + by + c = 0, (a \neq 0, b < 0)]$, which is:

$$u(a(x_i \pm w_i/2) + b(y_i \pm h_i/2) + c) \geq 0$$

where u=1, if the part above the line is cut, and u=-1if reversed.

For concave polygons, non-orthogonal boundaries at concave corners are treated as line segments. As Figure 2 shows, the above rules only apply within the boundaries of that line segment and combine the application of the subtraction rectangle.

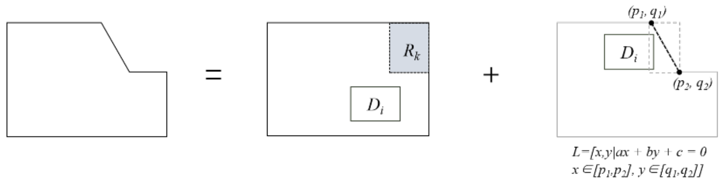


Figure2. Parse of the concave polygon boundary

Alternatively, a similar approach can be used to express the roads that traverse the site. Considering the width of the road and the setback requirements, the urban road is formulated as a line segment with width *d* and endpoints(p_1, q_1), (p_2, q_2), and the constraints are established to keep the corners of each parcel on the same side of the road and at a distance lager than $w/2$.

Figure 3. Generation tests for 10 parcels within three types of boundary cases

3.1.3. Natural boundary

Some urban construction sites are difficult to be parsed in the above two ways with non-linear internal and external boundaries, due to the fact that they are leaning on natural mountains and water bodies, or the need to preserve the natural landscape within the site during construction. These irregular boundaries can be expressed by AABB and unavailable regions in which sampling points are arranged at fixed intervals. The natural domain is decomposed into a point set of *n* sampled points, denoted as $\mathbb{P} = \{P_k\}$, where $P = [p, q, r]$and *r* is the control radius of point *P*.

Constraints are set to keep each point *k* in the set outside each parcel *i* and at a distance greater than the control radius:

$$(x_i - w_i/2) + (1 - \tau_{ij}^R)\mathrm{M} \geq p_k + r$$
$$(x_i + w_i/2) - (1 - \tau_{ij}^L)\mathrm{M} \geq p_k - r$$
$$(y_i - h_i/2) + (1 - \tau_{ij}^D)\mathrm{M} \geq q_k + r$$
$$(y_i + h_i/2) - (1 - \tau_{ij}^T)\mathrm{M} \geq q_k - r$$
$$\sum_{e \in \{T,D,L,R\}}^{n} \tau_{ij}^e \geq 1$$

where M is a large constant to ensure there is no overlap between parcel *i* and point *k* in direction e when $\tau_{ij}^e = 1$, and the inequality is always established when $\tau_{ij}^e = 0$.

3.2. INTERACTION WITH EXISTING ELEMENTS

The fixed elements on the site that affect the layout of the zoning, such as entrances, points of interest, roads, boundaries, axes, etc., are classified into point elements and linear elements as constant in the model (see Figure 4). There have been more studies on realising adjacency between rooms or parcels in previous studies, based on which this study considers the use of Euclidean distances to constrain the topology of the parcels and the existing elements of the site.

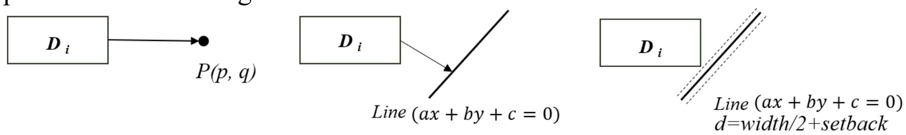


Figure 4. Interaction with fixed elements: points, boundaries, roads

Each point element is represented as it's horizontal and vertical coordinates by two constants, that is $P = [p, q]$. The distance and proximity of the parcel to the point is translated into the maximisation or minimisation of the objective of the Euclidean distance $(x_i - p)^2 + (y_i - q)^2$.

Each linear element is expressed using linear analytical formulas: $ax + by + c = 0$ and constants such as width and endpoint coordinates, and related to parcels by proximity, distance and adjacency. For proximity and distance, constraints are also established using the Euclidean distance from the parcel centre point to the line: $(ax_i + by_i + c)^2/(a^2 + b^2)$. While for adjacency, the distance between at least one corner of the parcel and the line should equal to w/2 on the basis of no overlap, the formula is expressed as:

$$-\mathrm{d} - \mathrm{M}(1 - \varepsilon_{ij}^R) \leq \mathrm{a}(x_i - w_i/2) + b(y_i - h_i/2) + c \leq d + M(1 - \varepsilon_{ij}^R)$$
$$-\mathrm{d} - \mathrm{M}(1 - \varepsilon_{ij}^L) \leq \mathrm{a}(x_i + w_i/2) + b(y_i - h_i/2) + c \leq d + M(1 - \varepsilon_{ij}^L)$$
$$-\mathrm{d} - \mathrm{M}(1 - \varepsilon_{ij}^D) \leq \mathrm{a}(x_i - w_i/2) + b(y_i + h_i/2) + c \leq d + M(1 - \varepsilon_{ij}^D)$$
$$-\mathrm{d} - \mathrm{M}(1 - \varepsilon_{ij}^T) \leq \mathrm{a}(x_i + w_i/2) + b(y_i + h_i/2) + c \leq d + M(1 - \varepsilon_{ij}^T)$$

$$\sum_{e\in\{R,L,D,T\}}^{n} \varepsilon_{ij}^{e} \geq 1$$

where M is a large constant to ensure the adjacency between parcel *i* and the road in direction e when $\varepsilon_{ij}^{e} = 1$, and the inequality is always established when $\varepsilon_{ij}^{e} = 0$.

3.3. FLEXIBLE ROAD NETWORK

There is always a mutual game between road and parcels in layout. This study achieves simultaneous optimisation of them by transforming roads into variables involved in layout. For common road networks, grid-type roads and circular roads are considered separately in the generation.

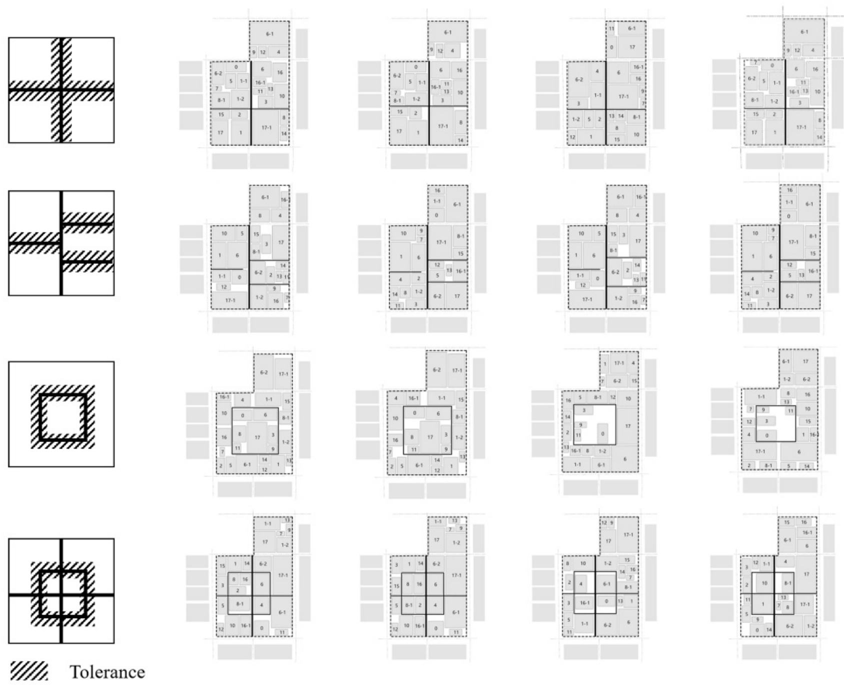


Figure 5. The layout generation with defined structure

In the rules of the grid-type layout structure, see the top two rows in Figure 5, the road network will be split into several horizontal and vertical roads with variable positions, creating variables for their positions and constraining the variable range. Under the base restriction that all parcels cannot cover roads, partitions and roads are generated simultaneously. In addition to this, it is possible, as with fixed linear elements, to constrain the relationship of parcels to variable roads, e.g. adjacent or only on one side.

Variable circular roads, see the bottom two rows in Figure 5, are constructed in a similar way as parcels but require the addition of the width attribute. On the basis that parcels do not overlap with roads, some topological constraints can be used to restrict parcels to being inside, outside, and adjacent to the road.

3.4. SOFTEN THE EDGE

Due to the logic of the cell construction logic, the parcels' boundaries in the layout results often fail to conform optimally to non-orthogonal boundaries. To address this limitation, a multiagent-based model is applied to "soften" the edges of parcels derived from the aforementioned layout results. Initially, circular entities with identical radii, are evenly distributed in each parcel with slight overlapping. Each entity is governed by objectives including non-overlap with others within the same parcel and maintaining a specified distance from those in neighbouring parcels. Additionally, they are constrained from extending beyond boundaries or crossing roads. After incorporating adjustments for roads and other inaccessible areas, the simulation can be started. The edges undergo dynamic adjustments through mutual collision among entities and collisions between entities and site elements. In this way, the topological relationships and overall areas of parcels remain relatively unchanged. (see Figure 6).

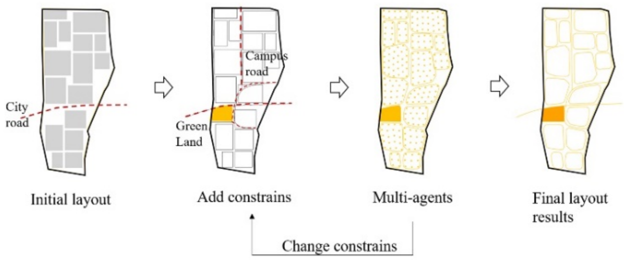


Figure 6. Soften the edge through circular muti-agents system.

4. User-configurable Layouts Generation

On the basis of the aforementioned algorithms, a complete user-configurable layout generation process is proposed as Figure 7:

First, data of site and zones is inputted and parsed. Site boundaries and environmental elements affecting the layout are automatically converted into specific structure which consists of AABB, abstraction rectangle, cutting line, exclusive points, and attraction elements consisting of points of attraction and information about roads. Zone data containing attributes of name, function, site area, density, plot ratio, number of storeys, etc is imported and can be divided into specific number of parcels that participates in layout.

Once the data has been inputted, specific constraints for layout can be defined by users, which includes the dimension constraints such as the aspect ratio and area share of each parcel, as well as the position constraints such as the typology between the parcels, and between the parcels and other elements. The soft constraints among them can be assigned different weights depending on the requirements and the results of the previous optimisation.

Furthermore, layout structures can also be defined, enabling structure-specific layout generation by setting up variable grid-type roads or ring roads. These custom constraints can be incrementally added and adjusted by feedback from layout results and generation efficiency, in order to determine the final constraint combination and obtain multiple optimisation results.

Finally, the boundary morphology of each parcel in the generated results is adjusted further using 3.4 approach, and the building data contained in each partition can be proportionally assigned to the sub-partitions and rendered in 3D.

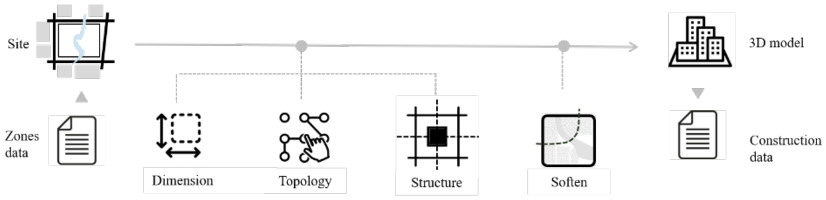
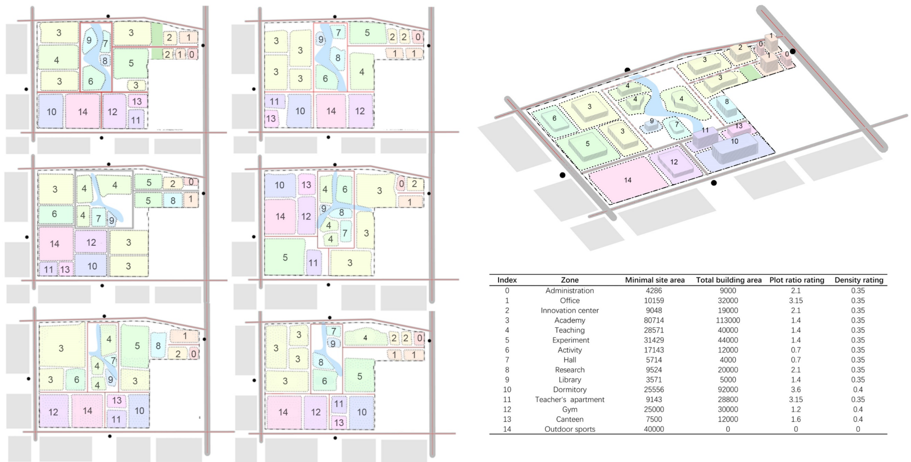


Figure 7. A complete layout generation workflow



Index	Zone	Minimal site area	Total building area	Plot ratio rating	Density rating
0	Administration	4286	9000	2.1	0.35
1	Office	10159	32000	3.15	0.35
2	Innovation center	9048	19000	2.1	0.35
3	Academy	80714	113000	1.4	0.35
4	Teaching	28571	40000	1.4	0.35
5	Experiment	31429	44000	1.4	0.35
6	Activity	17143	12000	0.7	0.35
7	Hall	5714	4000	0.7	0.35
8	Research	9524	20000	2.1	0.35
9	Library	3571	5000	1.4	0.35
10	Dormitory	25556	92000	3.6	0.4
11	Teacher's apartment	9143	28800	3.15	0.35
12	Gym	25000	30000	1.2	0.4
13	Canteen	7500	12000	1.6	0.4
14	Outdoor sports	40000	0	0	0

Figure 8. The application of this process to a campus layout

The main part of the program is implemented in C# and the construction and solution of MQP is done using Gurobi. The input and presentation of the geometry is implemented through Rhino platform with grasshopper. Figure 8 shows the application of this process to a campus layout with 15 function zones which divided into 20 parcels, where the data and layout requirements are based on the real project. The generation has reasonable running efficiency in the test: the gap decreases significantly in 20 seconds, and the solution remains stable after 50 seconds; there is no significant decrease in the efficiency of the solution when adding non-contradictory constraints.

5. Conclusion

The Quadratic Programming serves as a widely-used model to solve facility layout problem involving variable units for the feature of constraints and objectives. This paper extends the possibilities of applying the model to urban site layout, considering subjective design intents through a series of morphologically constrained algorithms. Experiments have shown that this approach can effectively satisfy hard conditions in the layout, achieve user-configurable constraints.

However, there are limitations inherent to the model. Firstly, each turn of generation is one-way and lacks visibility, limiting interactive engagement. Secondly, the construction principle of parcels limits the generation of layout with curved axes. Addressing the limitation could involve employing graphical methods to map layout domains prior to generation or optimizing morphology based on tensor field analysis post-generation. Additionally, integrating terrain considerations presents a significant challenge. Future research efforts will focus on developing a user-friendly interface allowing for configuration adjustments at each stage, and extending the framework to facilitate building layout generation within similar constraints.

References

Bao, F., Yan, D.-M., Mitra, N. J., & Wonka, P. (2013). Generating and exploring good building layouts. *ACM Transactions on Graphics*, 32(4), 122:1-122:10. https://doi.org/10.1145/2461912.2461977

Galle, P. (1981). An algorithm for exhaustive generation of building floor plans. *Commun. ACM*, 24(12), 813–825. https://doi.org/10.1145/358800.358804

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

Keatruangkamala, K., & Sinapiromsaran, K. (2005). Optimizing Architectural Layout Design via Mixed Integer Programming. In B. Martens & A. Brown (Eds.), *Computer Aided Architectural Design Futures 2005* (pp. 175–184). Springer Netherlands. https://doi.org/10.1007/1-4020-3698-1_16

Kropf, K. S. (2013). Ambiguity in the definition of built form. *Urban Morphology*. https://doi.org/10.51347/jum.v18i1.3995

Li, B., Guo, Z., & Ji, Y. (2015). Modeling and realizing generative design: A case study of the assignment of Ji Village. *Architectural Journal*, 05, 94–98.

Medjdoub, B., & Yannou, B. (2000). Separating topology and geometry in space planning. *Computer-Aided Design*, 32(1), 39–61. https://doi.org/10.1016/S0010-4485(99)00084-6

Peng, C.-H., Yang, Y.-L., & Wonka, P. (2014). Computing layouts with deformable templates. *ACM Transactions on Graphics*, 33(4), 99:1-99:11. https://doi.org/10.1145/2601097.2601164

Rhee, J., & Veloso, P. (2021). GENERATIVE DESIGN OF URBAN FABRICS USING DEEP LEARNING. *26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia: Online and Global, CAADRIA 2021*(pp. 31-40). The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA). https://doi.org/10.52842/conf.caadria.2021.1.031

Wu, W., Fan, L., Liu, L., & Wonka, P. (2018). MIQP-based layout design for building interiors. *Computer Graphics Forum*, 37. https://doi.org/10.1111/cgf.13380

Yang, Y.-L., Wang, J., Vouga, E., & Wonka, P. (2013). Urban pattern: Layout design by hierarchical domain splitting. *ACM Transactions on Graphics*, 32(6), 1–12. https://doi.org/10.1145/2508363.2508405