Maximizing Parking Space Utilization: An Algorithmic Approach for Architectural Plan Generation

Jiali Bao[1], Yuanjie Gao[2], Xinyi Zhao[2], Jichuan Yi[2], Lu Yin[3]

[1] China Intelligent Vehicle Innovation Platform
[2]Student, School of Architecture, Harbin Institute of Technology(Shenzhen)
[3] Senior Lecturer, School of Architecture, Harbin Institute of Technology(Shenzhen)

Abstract

Architectural plan generation, especially in the context of dynamic external outlines and internal obstacles, presents a significant challenge. This study introduces an algorithm to address this issue, using parking lot layout design as a case study. The approach combines Monte Carlo simulation and a greedy algorithm. We begin by subdividing complex polygonal outlines with internal obstacles into multiple simple rectangles. To maximize parking area utilization, the greedy algorithm generates rectangles with the largest possible area at each step. Each rectangle is created by integrating the Monte Carlo method, expanding from a randomly generated point horizontally and vertically until boundary contact, and repeating this process to identify the largest rectangle through 100 attempts. This continues until the remaining area is minimized, thus optimizing parking space usage. The simple rectangles are then categorized into perimeter and interior parking modes based on parking layout logic. Perimeter parking involves analyzing the convexity of angles formed with adjacent outlines to select one of eight layout schemes, while interior parking is based on predefined parking spot and lane dimensions. Quantitative and qualitative analysis are given to show efficiency of our proposal.

Keywords: Computational design; Parking Lot Plan Generation; Layout Optimization, Monte Carlo, Greedy Algorithm

1. Introduction

Floor plan layout is a fundamental task in architectural design, demanding significant human effort and time. Since the 1970s, researchers have explored generative methods to streamline this process(Mitchell Stiny 1978). However, most research in this area has focused on smaller spaces. This study shifts the focus to large spaces, particularly underground parking garages, which present unique challenges due to their extensive area, complex external contours, and various obstacles like machine rooms and staircases. The objective of this research is to develop a floor plan layout algorithm tailored to these specific conditions.

There are three main technical approaches to generative floor plan layout algorithms:

a.Shape Grammar: This method extracts the shape rules of objects and generates layouts in a manner similar to linguistic grammar. Developing an easy-to-use and versatile shape grammar system is challenging(Wu et al. 2019; Duarte 2005). However, this approach is not well-suited for underground parking garages.

b.Graph-based Machine Learning: This approach generates layout images covering various floor plan forms (Nauata 2020; Hu 2020; Gao 2023). It requires a large amount of high-quality floor plan data in a unified format, which is not always feasible. With limited data, the quality of generated layouts tends to be poor.

c.Optimization: This method is computationally efficient and fast, making it suitable for assisting in the design phase of architectural projects. There are already numerous successful applications, such as using grid-based calculations to solve the layout of residential units and classroom buildings (Veloso et al. 2018; Michalek et al. 2002; Elezkurtaj and Franck 2020; Verma and Thakur 2010). This research adopts this technical route to explore specific optimization algorithms most suitable for the floor plan layout of underground parking garages.

By focusing on the optimization approach, this study aims to identify the most effective strategies for addressing the unique challenges posed by large underground parking garages. The goal is to create a layout algorithm that maximizes space efficiency while ensuring functionality and ease of use.

Contact Author: Lu Yin, Senior Lecturer, School of Architecture, Harbin Institute of Technology(Shenzhen),
Address: Building T4, HIT Campus, The University Town of Shenzhen, Harbin Institute of Technology(Shenzhen), Shenzhen City, Guangdong Province, China
Tel: (+86)18645959745
e-mail: yinlu2021@hit.edu.cn

2. General Principles

The algorithm developed in this study aims to be a quick and convenient tool for architects during the conceptual design phase. Its primary purpose is to help rapidly estimate the total parking capacity of an underground garage based on the layout of above-ground buildings. Drawing from practical engineering experience in underground garage design, the following principles must be adhered to for the garage layout:

a. **Flexible Contour Handling**: The external contour of the garage is often irregular, influenced by site boundaries and the design of above-ground buildings and landscapes. The target algorithm should be capable of flexibly adapting to changes in the contour edges.

b. **Adjustable Internal Obstacles**: Core structures of the above-ground buildings, such as core tubes, equipment rooms, and staircases, are considered obstacles within the garage. The target algorithm should allow for flexible adjustment of the size and position of these obstacles.

c. **Efficient Parking Arrangement**: The internal layout should primarily consist of regular horizontal or vertical parking spaces. The dimensions of the parking spaces should be 2.4m x 5.3m, with aisle widths of at least 5.5m. The placement of columns should prioritize maximizing the number of parking spaces.

d. **Interruption**: To minimize the need for backtracking and to improve visibility of available parking spaces, vertically aligned parking spaces should be interrupted by an aisle approximately every 40 meters.

The specific dimension requirements are shown in Table 1.

Table 1. Size Principles for Parking Lot Dimensions

Configuration	Value
Parking Slot Length	5.2 (m)
Parking Slot Width	2.4 (m)
Road Width	5.5 (m)
Vehicles num in Group (horizontal)	5
Vehicles num in Group(vertical)	2
Pillar Width	0.4 (m)

3. Methods

Our proposal for parking generation consists of two main stages: area division and parking slot generation, as illustrated in Fig. 1. Initially, the parking layout and relevant parameters, such as parking slot width, length, etc., are digitized to form a model. There are three following stages. In stage 1, parking slots are generated along the outer borders of the parking area as well as within the divided rectangular areas. In stage 2, the parking area is divided into the largest possible rectangles to maximize the efficiency of parking slot arrangement. This ensures optimal use of space and facilitates easier layout planning. In stage 3, parking slots are generated inside the sub divisions from stage 2.

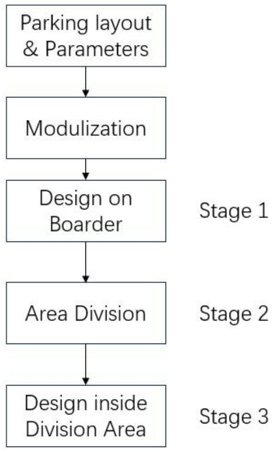

Fig.1. Main frame of parking generation

3.1. Stage 1 Outer layout

For the outermost perimeter of the parking lot, parking spaces are arranged in a grid layout to maximize space utilization. At the edges and corners, non-overlapping rules are set to ensure smooth entry and exit of vehicles. The arrangement logic for the outer row of parking spaces is illustrated in Fig. 2.

The key focus for the outermost perimeter is the corners. For protruding points, both sides must be clear of obstructions (see Fig. 2). For indented points, to accommodate more parking spaces, parking spaces on one side may extend over the template width of the opposite side. The specific rules are depicted in the figure.

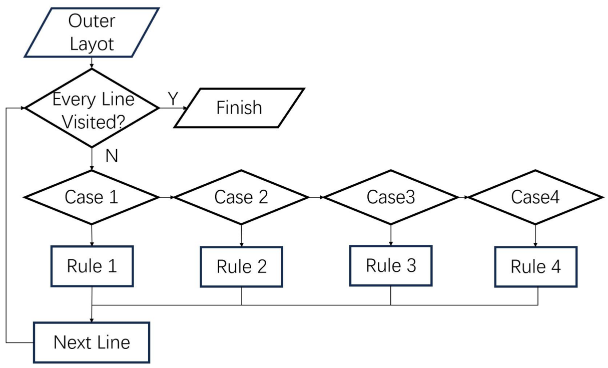

Fig.2. Flow chart of Outer Layout Generation

There are 4 possible patterns of edge length, as shown in Table 2 and Fig.3. The pattern is decided by the start point and the end point, which leads to the different edge length.

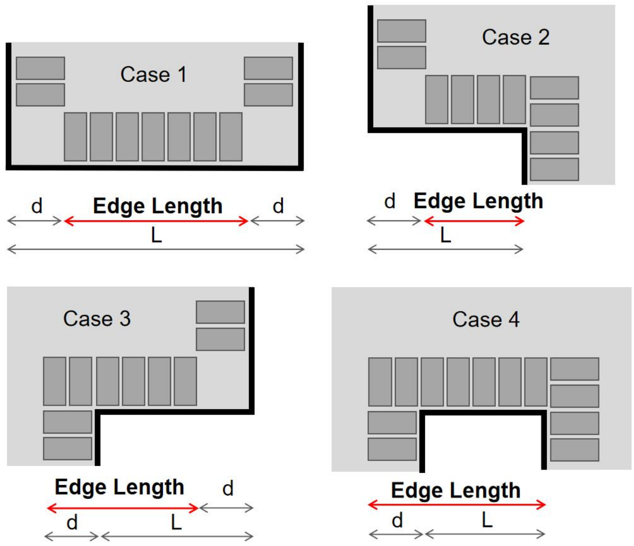

Fig.3. Possible Patterns of Edge Length

Table 2. Possible Patterns of Edge Length

Case	Start Point	End Point	Edge Length
1	Convex	Convex	L-2d
2	Convex	Concave	L-d
3	Concave	Convex	L
4	Concave	Concave	L+d

3.2. Stage 2 Area Division

The parking area is typically an irregular polygon, necessitating its division into large rectangles for efficient parking slot arrangement. An example is given as Fig. 4.

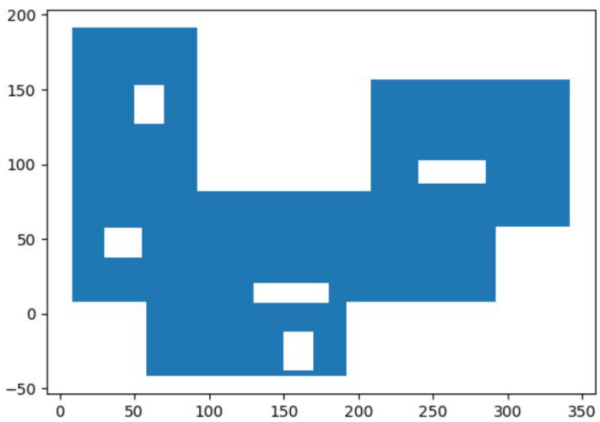

Fig.4. Example of Typical Parking Area

The area division process utilizes a Monte Carlo algorithm and a greedy algorithm, as shown in Fig. 5.

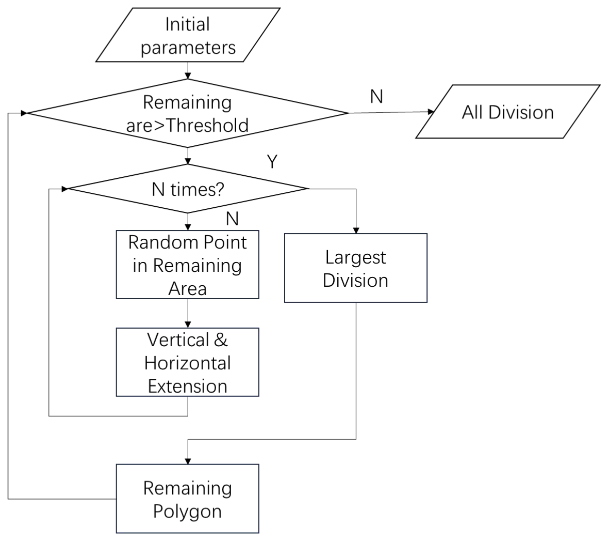

Fig.5. Flow chart of area division generation

In each iteration, a random point within the parking area is selected, from which vertical and horizontal extensions are made until the border is hit, forming a rectangle. This process is repeated n times per iteration, recording and cutting the largest resulting rectangle from the parking area. The algorithm is reapplied to the remaining area to generate the next largest rectangle, as illustrated in Fig.s 5.

An example of area division is given in Fig.6, showing that No.1 area is the largest, and the No.2 area is the second largest.

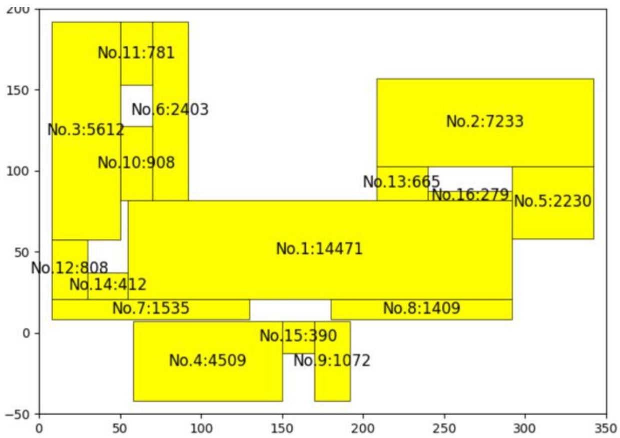

Fig.6. Example of Area Division Result

3.3. Stage 3 Parking Slot Generation

Finally, parking slots are generated within all the divided areas according to the configuration shown in Table 1. The goal is to arrange as many parking slots as possible to fully utilize the available space. The final layout is illustrated in Fig. 7.

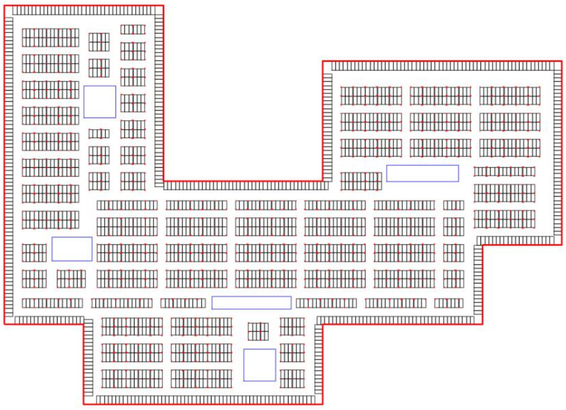

Fig.7. Example of Parking Generation Result

4. Analysis

To evaluate the performance of our proposal, we implemented the algorithm on three layouts of real parking areas with total area of about 1000~2000 ㎡. . The original images and generated results are compared in Fig. 8, Fig. 9 and Fig. 10. For simplicity, obstacles and local contours were simplified to squares, shown in blue.

We calculated the average space needed per parking slot using Equation (1):

$$\text{Parking Lot Area per Car} = \frac{\text{Total Area of Parking Lot}}{\text{Total Number of Parking Slots}} \quad (1)$$

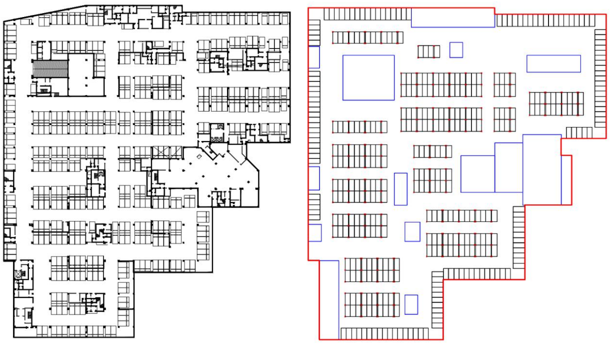

Fig.8. Comparison of Actual Case 1 and Generated Layout Plan

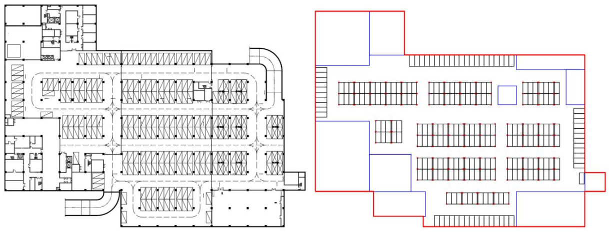

Fig.9. Comparison of Actual Case 2 and Generated Layout Plan

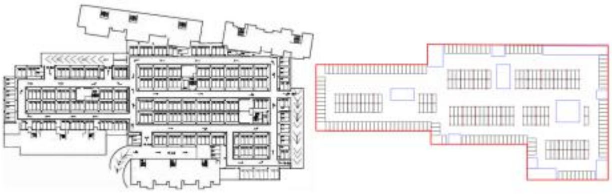

Fig.10. Comparison of Actual Case 3 and Generated Layout Plan

Based on the comparisons of the actual cases and generated layout plans, the following analysis can be derived from Table 3:

a. Layout: From the results of the three case studies, we observe that although our generated layouts differ somewhat from the actual layouts, there are many similarities in the overall structure. This indicates that the algorithm has the capability to simulate the layout thinking of designers to a certain extent.

b. Efficiency: We compared the total number of parking spaces in each actual case with our generated results. We also calculated the average parking lot area required per vehicle. The results show that the total number of generated parking spaces is slightly higher than in the actual cases, indicating higher parking efficiency per vehicle. While actual layouts need to consider more specific issues and cannot be as ideal as the generated layouts, we also calculated the ratio of total generated parking spaces to the total parking spaces in the actual cases. This ratio consistently remains around 1.1, suggesting that the results obtained using this algorithm are valuable for estimating the actual number of parking spaces.

c. Time Complexity: We recorded the computation time for each calculation, which ranged from 4 to 11 seconds, on a . This demonstrates that the algorithm is quick and efficient in its operations and adjustments.

Table 3. Comparison of Actual Case 1,2&3 and Our Generated Layout Plan

Case	Total Area of Parking Lot (㎡)	Total Parking Slots Number in Actual Case	Parking Lot Area per Car in Actual Case(㎡)	Total Parking Slots Number in Our Case	Parking Lot Area per Car in Our Case(㎡)	Ratio of parking slots number to actual cases	Time Consumption(s)
1	17817	392	45.45	424	42.02	1.08	10.9
2	11121	223	49.87	250	44.48	1.12	4.9
3	9813	260	37.84	281	34.92	1.08	6.2

5. Conclusions

In this paper, we introduced a three-stage algorithm to automatically generate parking space layouts. Our proposal demonstrated high efficiency with real data, yielding results similar to handcrafted designs.

Using the scenario of an underground parking lot, we proposed a floor plan generation algorithm tailored for large areas with irregular contours and internal obstacles. This algorithm, based on Monte Carlo and greedy methods, provides an optimized layout solution quickly when given the coordinates of the external contour and internal obstacles. By comparing the layout generated by this algorithm with actual engineering case layouts, we demonstrate its reliability.

Although the current results of this algorithm do not reach the precision required for final floor plans, it significantly aids architects in the early stages of design. When adjustments to the garage contour, staircases, ramps, and other elements are needed, architects can quickly determine the feasibility of these changes without manually placing parking spaces to verify parking metrics each time. This greatly saves designers' time and effort.

In future research, we aim to enhance the accuracy and general applicability of the floor plan calculations further. Additionally, we will explore incorporating design elements from other related disciplines, such as building services, to achieve a more precise and comprehensive construction drawing design through the algorithm.

6. References

1) Mitchell S. (1978) The Palladian grammar. Environment and Planning B, 5, 5-18.
2) Michalek, J., Choudhary, R., and Papalambros, P. (2002) Architectural layout design optimization. Engineering optimization, 34(5), 461-484.
3) Elezkurtaj, T., and Franck, G. (2002) Algorithmic support of creative architectural design. Organization, 2, 16.
4) Duarte, J. P. (2005) A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira. Automation in construction, 14(2), 265-275.
5) Verma, M., and Thakur, M. K. (2010) Architectural space planning using genetic algorithms. The 2nd International Conference on Computer and Automation Engineering (ICCAE) 2, 268-275.
6) Veloso, P., Celani, G., and Scheeren, R. (2018) From the generation of layouts to the production of construction documents: An application in the customization of apartment plans. Automation in Construction, 96, 224-235.
7) Wu, W., Fu, X. M., Tang, R., Wang, Y., Qi, Y. H., and Liu, L. (2019) Data-driven interior plan generation for residential buildings. ACM Transactions on Graphics (TOG), 38(6), 1-12.
8) Nauata, N., Chang, K. H., Cheng, C. Y., Mori, G., and Furukawa, Y.(2020) House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23 – 28, 2020, Proceedings, I(16), 162-177.
9) Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., and Huang, H. (2020) Graph2plan: Learning floorplan generation from layout graphs. ACM Transactions on Graphics (TOG), 39(4), 118-1.
10) Gao, X., Guo, X., and Lo, T. (2023) M-StruGAN: An Automatic 2D-Plan Generation System under Mixed Structural Constraints for Homestays. Sustainability, 15(9): 7126.