Research on the Spatial Layout Design of University Educational Buildings Based on Rule Screening and Multi-agent System

Yixuan Zeng[1], Qiaoming Deng[2], and Yubo Liu[2,3](✉)

[1] College of Architecture and Urban Planning, Tongji University, 1239 Siping Rd, Shanghai, China
[2] School of Architecture, South China University of Technology, Guangzhou, China
liuyubo@scut.edu.cn
[3] State Key Laboratory of Subtropical Building Science, School of Architecture, South China University of Technology, Guangzhou, China

Abstract. Unlike traditional empiricism-based building design, the data-oriented quantitative analysis method is more rigorous and intuitive, taking into account a variety of factors such as site conditions, functional requirements and design specifications, and combining computer technology to propose a more rational and efficient design strategy. This study takes the design logic process and algorithm rule screening as the entry point to explore the design method of using multi-agent body algorithm planning to realize the spatial layout of university education buildings. Based on multi-agent algorithms, spatially rich and morphologically complex architectural solutions can be quickly generated, and new designs with generality and universality can be produced by changing the initial shape and syntax rules. The authors attempt to design a program based on a multi-agent body system, where architects only need to set initial parameters to quickly construct a variety of initial volume solutions, offering a wide range of possibilities for initial design.

Keywords: Multi-agent · Self-organization · Bottom-up · Generative design · Educational buildings

1 Introduction

The design of a university is a large-scale urban design for small education, which is realized for such a combination of buildings, a multi-body structure or a multi-faceted design, as well as a multi-body design and multiple decent design perspectives. Teachers need to spend more time participating in the whole process of design.

Computer-aided design has been used in the field of architectural design maturely, such as auxiliary drawing, modeling, statistics and so on. These auxiliary tools can not only help designers better improve the model, but also combine with other related disciplines to complete the full-cycle design of construction projects, so that various disciplines can better coordinate in actual projects. With computational design, the designer

P. F. Yuan et al. (eds.), *Hybrid Intelligence*, Computational Design and Robotic Fabrication,
https://doi.org/10.1007/978-981-19-8637-6_5

only determines the general goals of the design, and automatically solves it by setting a series of generative rules and generative logic. Multi-agent is a bottom-up approach to model building. In the multi-agent method, the design will gradually spread out with the interaction effect between the elements, and can ensure that each result is the evolution of the previous result, and can ensure that the design process is bottom-up and evidence-based. Based on the self-generating theory of multi-agent systems, the method of generating educational buildings in colleges and universities is explored by means of rule screening and algorithms. Designers only need to adjust input conditions or parameters to obtain various design results in a short time. This paper attempts to explore the use of computational design thinking to solve campus architectural design problems, and to tap the potential of computational generative design in architectural design.

2 Discussion on the Generation Method Based on Multi-agent

The concept of the multi-agent model was extended and evolved from the study of cellular automata. In 1940, John von Neumann and Stanislaw Ulam proposed a device for self-reconstruction and regeneration [1]; in 1970, John Conway simplified their vision and designed a “survival game” [2]. The color of the squares is used to indicate the “survival” or “death” of the cells. After determining the number of grids, the proportion of initially generated cells, and the rules of cell change, each unit cell will adjust its state according to the state of its neighbors. The system eventually reaches equilibrium.

Multi-agent system is a computing system composed of institutions that consists of multiple agents interacting in an environment. Its concept was first proposed by Minsky at MIT in the 1980s [3]. A multi-agent is a dynamic system in which all internal intelligences interact and interact with the environment. The multi-agent system focuses on the interaction, cooperation and conflict between multiple intelligences, emphasizing the group nature of intelligence rather than individual intelligence. The application research of multi-agent system in architecture mainly focuses on urban morphogenesis, human flow simulation and functional topological relationship generation. The ETHZ-CAAD laboratory has completed a practical project in the station square in Groningen, the Netherlands, with a multi-agent agent acting as a pillar being generated by a column grid arrangement of a planar non-orthogonal system [4]. The team of Li Biao developed the “CUBE101” simulation agent’s life and death game to generate the floor plan and spatial form of the collective housing [4]. Guo Zifeng from Southeast University studied multi-agent systems constrained by functional topological relations, explored functional topological multi-agent systems in three-dimensional space, and realized the layout of three-dimensional functional space [5].

Architecture is a very complex subject, involving many influencing factors, and there is no clear judgment and evaluation standard for its results. Based on the multi-agent system, under the premise of the same number of input rules, as long as the evaluation criteria and goals are set, the best results can be obtained under the input conditions. At present, there are few researches on the generation design of educational building layout based on multi-agent, and research can be carried out on issues such as building spatial layout, functional topological relationship, and building group generation. At the same time, the bottom-up generative design idea of multi-agent system is an effective way to solve the layout and functional space layout of educational buildings. Time-consuming issues under some specific conditions. Although the multi-agent system generates building plans, it is still necessary to split the design steps, perform program calculations in batches, and perform manual intervention on the results in time to achieve results that meet the set expectations. But it is undeniable that the potential of multi-agent systems in architectural design has not been fully tapped.

3 The Rule Deduction of Space Combination

3.1 Modular Unit Design

The author studies and analyzes the main space of common teaching activities in colleges and universities, such as ordinary classrooms, amphitheatre classrooms, book reading rooms and laboratories. These regular teaching spaces provide students with a place to study and are also important places for students to carry out public activities and daily interactions. Analyze the basic dimensions of these common functional spaces, take the 8-m column span as the basic modulus, and take the 3 × 3 modular standard unit as the basis, and organize the main teaching space into three modular units of 1 × 1, 2 × 1, 2 × 2, and 2 × 3.

3.2 Unit Plane Translation and Growth Rules

In the 3 × 3 modulus unit, the author divides the path, and uses this path division as the main plane generation in the future. The path division method can be expressed by a program algorithm. We determine the position of the path by the variables n and m, and denote the side length of the module as quantitative mod, then the vector of the X-axis is X(1,0,0), and the vector of the Y-axis is Y(0,1,0), the position of the initial point is determined by the above parameters (Fig. 1).

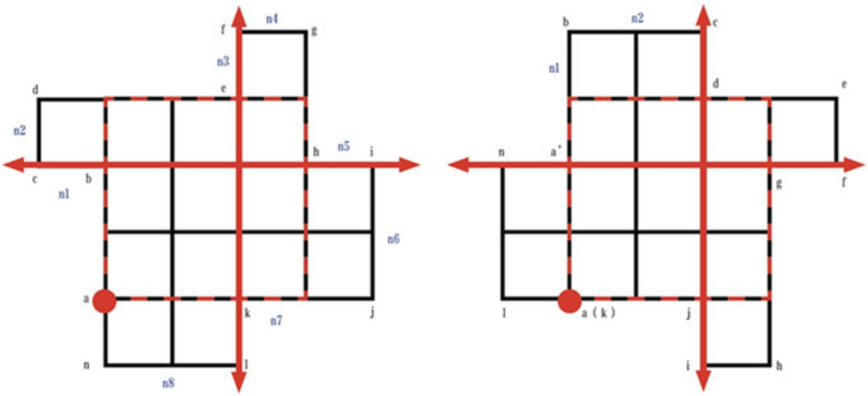

Fig. 1. 3 × 3 modulus unit growth rule (drawn by the author)

A simple modular unit can be derived from a variety of plane generation schemes according to three growth rules. The first rule is the number of changes. Depending on the way the path is divided, the core module can grow into different branches. The second rule is the direction change. The growing branches are generated in a clockwise or counterclockwise direction, and different growth directions determine the next generation law. The third rule is step size. According to the different random seeds, the step size modulus of the length and width of each branch is determined. This rule is the main factor that affects plane richness. The way the module grows rules can be expressed in a program. The following four growth directions are used as examples to illustrate, and the same is true for other directions (Fig. 2). Select an initial control point a, deduce the step length from this point to the X-axis/Y-axis, generate the next point, and repeat in turn, until the enclosing and generate a single plane, and thus obtain the path, path intersection and port, and port direction.

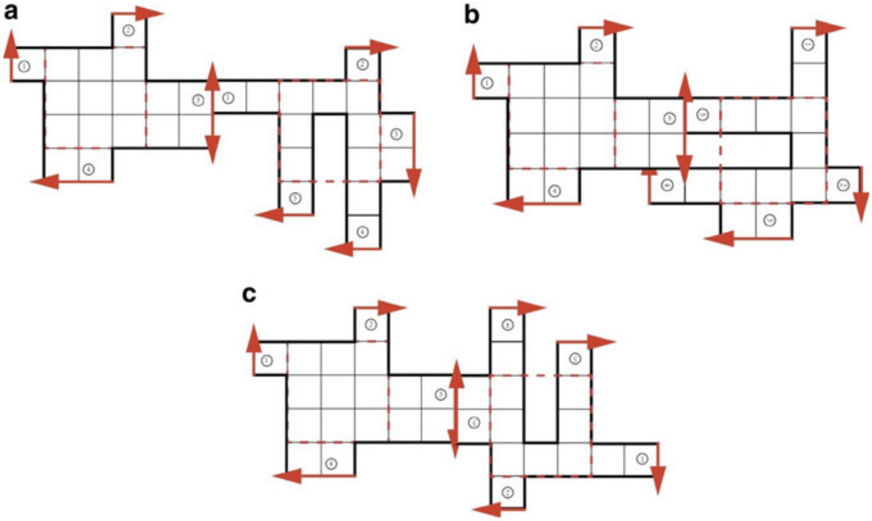

Fig. 2. (**a**) Directly connected when the ports are parallel. (**b**) Directly connected when the ports are vertical (**c**)

3.3 Unit Connection Rules and Translation

Rotation rule: After the monomers are generated by the program algorithm, according to the relationship between the ports of each monomer and the directions of the ports, the monomers are spliced in pairs (Fig. 2). When the angle between the ports of the monomer A and the monomer B is 180°, the rule of direct connection is satisfied. When the angle between the ports of the monomer A and the monomer B is 90° or 270°, the monomer B is clockwise or Select 90° counterclockwise to connect. When the angle between the ports of monomer A and the monomer is 180°, the monomer B rotates 180° to connect. Through these three rules, the principle that the planes of the monomers are connected can be satisfied.

Connection compensation rule: When the four monomers are connected to each other and exist in the form of a closed loop, due to the randomness of shape generation, it cannot be ensured that the shape finally exists in an enclosed form. Therefore, a connection compensation rule is added (Fig. 3). The stabilized form will retrieve the surrounding blocks of the interface. When the surrounding blocks overlap or border, the interface will automatically close, and when the surrounding blocks do not have the above situation, it will remain open.

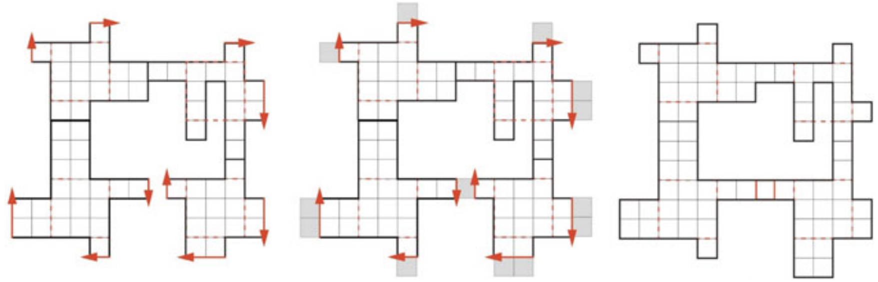

Fig. 3. Monomer connection compensation rules: detect the blocks around the interface, if they overlap or border, the blocks are connected

Overlapping and merging rules: When four monomers are connected to each other and exist in the form of a closed loop, due to the randomness of shape generation and the final positional relationship of each shape, there may be overlapping between monomers (Fig. 4). Therefore, overlapping merge rules were added. The stabilized shape will search for the overlapping area between the monomers. If the overlapping unit modules are less than or equal to 2 units, they will be automatically combined into a complete shape, but if this requirement is not met and there are too many overlapping units, this scheme will The generation is invalid, and a new scheme is recalculated.

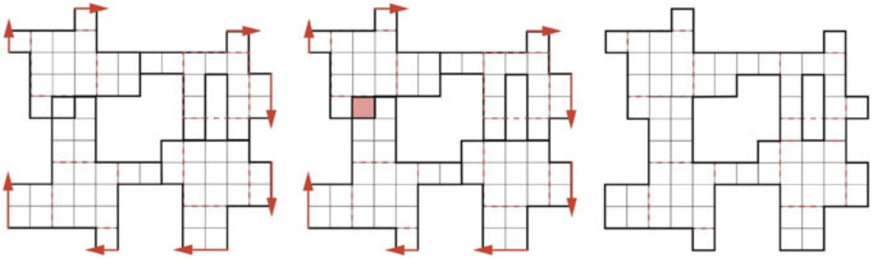

Fig. 4. Monomer overlapping merge rules: detect overlapping areas, if less than 2 units, generate a flat plan

3.4 Core Tube and Layout

Auxiliary modules (toilet and traffic space) are placed in each 3 × 3 core, the location is determined by the location of the intersection of the paths. Auxiliary modules are added at the connection of the monomers, and at the same time, the fire protection distance requirements are met. The distance between the fire stairs in the strip corridor should not be greater than 40 m (5 module units), and the space at the end of the fire stairs should not be greater than 22 m (3 module unit). After the path direction and the position of the core tube are determined, the basic functional space is divided, and the large space modules are placed on the plane first, and then the small space modules are placed, and the aisles are kept connected to each other (Fig. 5).

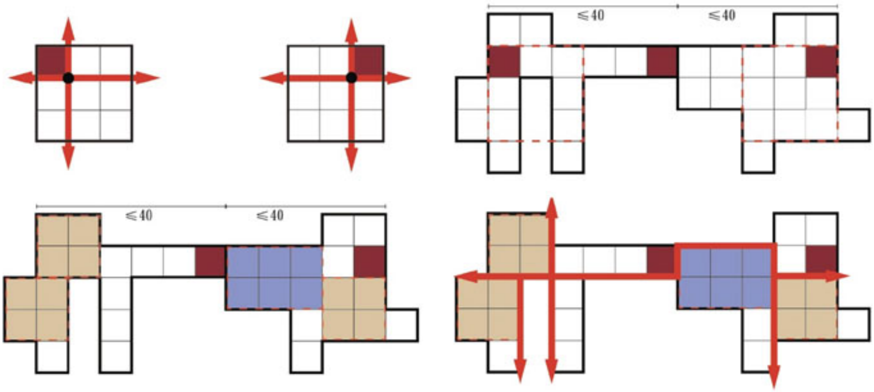

Fig. 5. Core tube and layout rules

4 Exploration of Space Composition and Scheme Comparison

4.1 Program Design Ideas

Based on a campus site in the south, this research combines the design process with multi-agent algorithm technology, translates the traditional design links into computer programs, and realizes the generation of university building space layout based on multi-agent system. The main work links of this experiment are as follows: First, the site element processing, the site is subjected to vectorized grid processing and digital output, the site is assigned grid cells and the complete site grid is screened out. Second, set the initial parameters, through the basic parameter control terminal, set the initial point position, random seed number, and arrangement rules. Third, agent translation and operation rule setting, set different agent arrangement forms and data attributes, obtain the plane layout, path, and core position of each layer, and form the final layer of each layer according to the compensation and merging rules. Architectural Design. Fourth, generate a comprehensive score for the scheme, and select the appropriate scheme based on the objective basis of plot ratio, boundary richness, block richness, and comprehensive index. Fifth, based on the rationality of the spatial functional layout, optimize the block relationship and functional layout for the optimal solution.

4.2 Evaluation Index Factors

Based on the grasshopper platform, the above rules are simplified into four basic battery packs: initial battery module, interface direction battery, initial connector battery, and process connector battery. Only need to set the initial plane generation position, adjust the parameters as needed, the positional relationship between the monomers and the connection between the monomers will change, and a variety of changeable layout forms will be generated. Taking the side length of 216 × 216 m as the simulation site, the intelligent unit is placed in the simulation site for calculation. After many tests, four types of generative layouts have been concluded, namely, the whole is centered and not out of bounds; the whole is scattered and not out of bounds; the center is arranged and out of bounds, and the whole is scattered and out of bounds (Fig. 6). These four types of randomly generated layout schemes are difficult to judge by human judgment. The richness and standardization are the goals pursued, and they are refined into floor area ratio, boundary richness, volume richness, and transboundary coefficient., the overlap coefficient of these five basic indicators.

A (Floor area ratio) is the ratio of the plane area of the generated building to the plane area of the site, and this ratio determines the accommodation density relationship of the site. B (Boundary richness) is the ratio of the sum of the perimeters of the generated plane to the sum of the perimeters of the 3 × 3 core space plane. This ratio determines whether the scheme is stretched or intensive, scattered or centered. C (block richness) is the ratio of the overhead or setback area of each floor of the generated plane to the area of the floor. This ratio determines the spatial experience of the scheme. When a scheme has more overhead or setback space, a certain To a certain extent, it can be explained that the scheme has better space tour and building roaming experience. D (out-of-bounds coefficient) is to judge whether the generated plane exceeds the building red line. This value determines whether the generated plan is reasonable. When the building does not exceed the red line, the value is 1, otherwise, the value is 0. E (overlap coefficient) is to judge the overlap relationship between randomly generated planes. Due to the randomness of the building plane generated by the algorithm, the location of the building units at each step is uncertain, so there is a situation where the building units overlap each other. This value determines whether the generated scheme exists or not. Set a comprehensive mean Score, the formula is

$$Score = (a + b + c)^{*} d^{*} e \tag{1}$$

The larger the value of Score, the better the performance of each index of the scheme, and the better it can meet the design requirements of richness.

4.3 Comparison and Selection of a Campus Design Scheme

After extracting the vector boundary of a university site in the south, a grid with a modulus of 8 is built into the site, and a basic grid frame is divided according to the road network

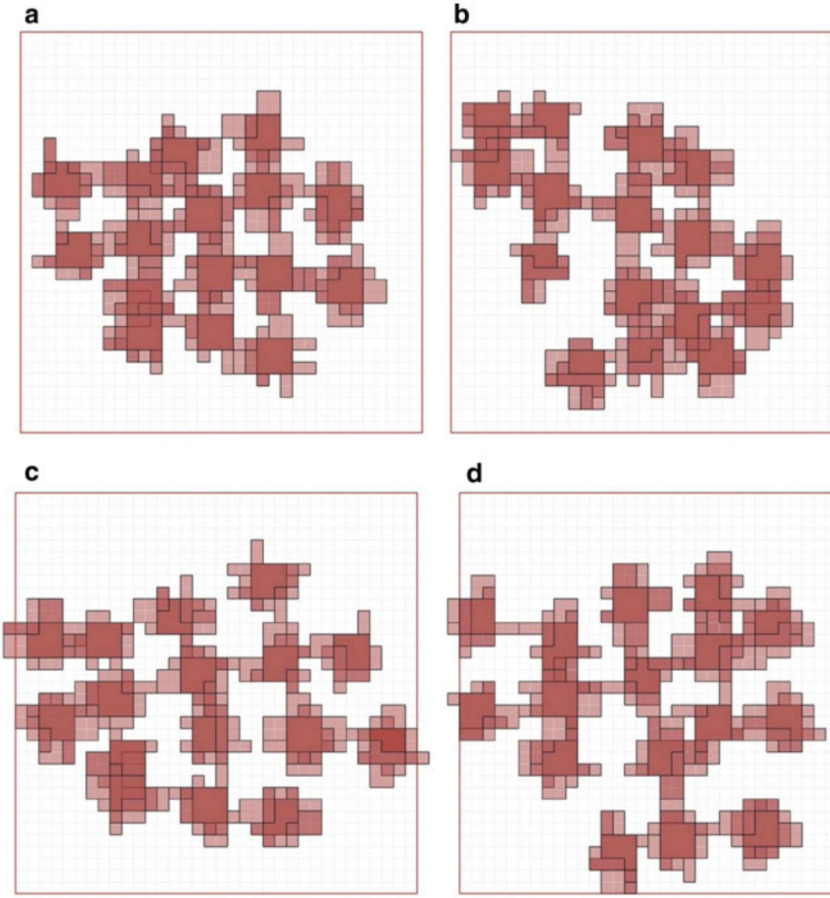

Fig. 6. Four layout types: **a** the whole is centered and not out of bounds; **b** the whole is scattered and not out of bounds; **c** the center is arranged and out of bounds; **d** the whole is scattered and out of bounds

relationship, and a point in the lower left corner is selected as the initial point for shape generation (Fig. 7). The architect only needs to set the initial parameters in the program, and then the design scheme can be generated, which provides more possibilities for the initial architectural design. Among them, the quantitative parameters are: site boundary A, grid modulus B, control grid C, and the variable parameters are: control point U, random parameter V, generation method W. These factors determine the final mass generation.

The author divides the generation schemes into three categories: single arrangement, staggered arrangement, and forward and reverse arrangement. Monolithic layout, that is, each unit is generated by a control point, each building volume is like an independent small house, and the position between each unit is determined by the initial point. The multi-storey building layout has been spatially designed to The first floor is the benchmark, and 0 ~ 2 step units are gradually retreated upward to ensure that the building

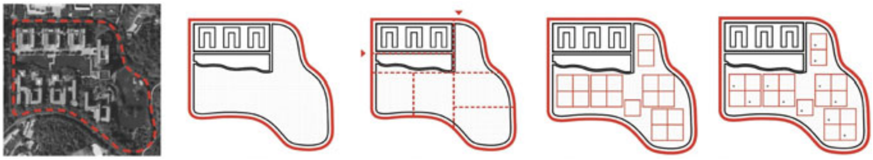

Fig. 7. Site meshing and setting initial points

space on each floor has more dislocations, and the building space on each floor can get better lighting. However, the connection between the monomers is not strong, the enclosures between the monomers are arranged in a "well" shape, and the enclosure relationship is relatively monotonous (Fig. 8).

Fig. 8. Single arrangement: layer by layer to get a single building

Staggered arrangement, that is, the building volumes on each floor, like building blocks, are staggered in the order of "horizontal-vertical-horizontal-vertical". A control point can determine the arrangement of 2 unit blocks. The spatial layout of the multi-storey building is staggered horizontally and vertically, and 0~2 step units are retreated up by layer to ensure the dislocation relationship and lighting of the building space on each floor. The generated architectural plan has a relatively close connection in the horizontal and vertical layout, showing rich features like "weaving" (Fig. 9).

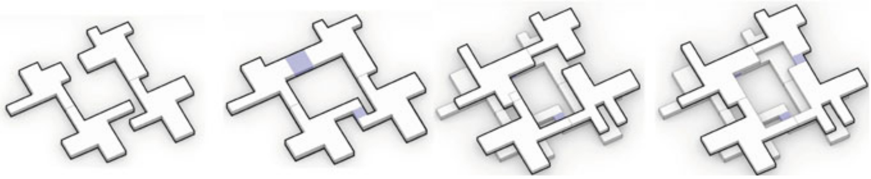

Fig. 9. Staggered arrangement: criss-cross arrangement of blocks

The forward and reverse arrangement, that is, the building volume of each floor, like a greedy snake, is enclosed in a clockwise direction, and then in a counterclockwise direction, and so on. One control point can determine the arrangement of 4 unit blocks, and only 5 control points are needed to complete the layout of the entire site. Based on the basic rule of "step back and stagger each other", each floor is set back 0 ~ 2 step units, ensuring the horizontal and vertical staggered arrangement of the multi-storey building space. The connection between the four monomers is very close. As long as the initial building block changes, the enclosing relationship of the entire building will also change. The generated shape not only has the characteristics of criss-cross texture, but also reflects the "maze". layout features (Fig. 10).

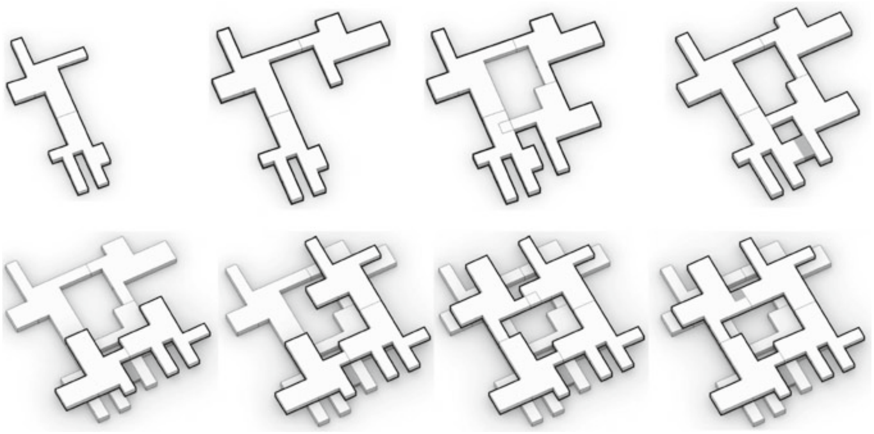

Fig. 10. Forward and reverse arrangement: arrange blocks in a clockwise-counterclockwise direction

Based on the annealing algorithm of galapagos, the author obtains the optimal solution of weighted value through multiple iterations, counts various numerical indicators in the experimental process, and obtains the optimal solution of weighted value for three generation methods. Then, the numerical data of the optimal solutions of the three schemes are sorted and summarized, and a horizontal comparative analysis is carried out (Fig. 11). In order to make a more comprehensive evaluation, the author organizes the obtained data and summarizes it into five indicators: plot ratio, boundary richness, block richness, open space value, open value, and comprehensive value. It can be seen from the data results that the forward and reverse scheme is higher than the other two schemes in terms of boundary degree and block richness, indicating that the generated scheme has better spatial effect. At the same time, in terms of volume ratio, it is the same as the staggered type, and is much larger than the single type. However, its vacant land value is small and its openness is high, which means that the site utilization rate is high, the volume is richer, and the overall performance is better. In the analysis of physical environment performance, the lighting performance is also worthy of appreciation. Therefore, the optimal solution of the forward and reverse can be used as the final detailed design scheme (Table 1).

5 Strengths and Weaknesses

Based on the self-organization characteristics and "bottom-up" logic characteristics of multi-agent systems, this research quantifies constraints such as various normative regulations and people's subjective orientations, and translates buildings into systems that can operate within the system, It is an agent that restricts and interacts with each other,

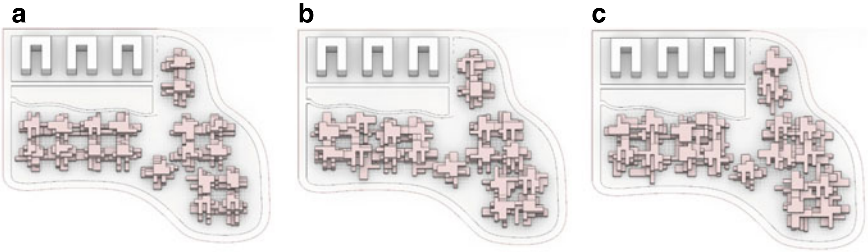

Fig. 11. Comparative analysis of three methods, **a** Single arrangement; **b** Staggered arrangement; **c** Forward and reverse arrangement

Table 1. Indicators for the three programmes

	Single	Staggered	Forward and reverse
Volume rate	1.24	1.15	1.13
Boundary	2.22	2.33	2.24
Bulkiness	4.51	3.79	3.31
Openness	3.44	3.06	3.12
Comprehensive	4.48	4.14	3.61

and can interact with the environment, so as to complete the spatial layout of college campuses. At the same time, this is a universal architectural design generation system. The design does not emphasize the location and surrounding environment of the site, but pays more attention to the richness of its building volume.

Compared with traditional architectural design, the multi-agent system used can obtain endless solutions, avoiding the continuous attempts of manpower for a large number of solutions, which can greatly improve the generation efficiency, expand the possibility of forms, and make it easier to find and screen. Better solution. It provides a large number of options and possibilities for architects in the early stage of sketch design, which is conducive to enriching the architect's design material library and stimulating the designer's creative potential. At the same time, the generated building group is rich in volume and has a strong sense of spatial hierarchy. While doing daily scientific research and study, students can also stroll through the building space and relax. These rich and interesting building volumes will become places for students to communicate and communicate.

To summarize the process and results of this research, there are still many imperfections. At the programming level, this research realizes the multi-agent system experiment and volume generation based on Grasshopper and Python programming languages, but the functional plane deepening is redrawn in Revi, and cross-platform operation will cause the problem of model information conversion, and The program cannot achieve the effect of real-time linkage on multiple platforms. At the same time, because the program code behind the model is not concise enough, it takes a long time to generate the optimal solution. At the level of interaction design, this program code does not have a concise interactive interface and clear operation instructions, and cannot be directly oriented to the underlying design workers without programming foundation. Furthermore, the scoring data cannot be converted into standard charts for display in real time, and it is impossible to intuitively analyze and interpret the generated design scheme rationally. At the architectural design level, the building volume automatically generated by the program is too staggered, resulting in misalignment of the upper and lower column grids, dark rooms, and poor building lighting. This reduces the richness of algorithmically generated architectural solutions and goes against the original pursuit of algorithmically generated design.

6 Future Outlook

The generation system based on multi-agent and rule screening does not yet have the ability to self-optimize, and the future optimization space will consider adding core methods such as machine learning to make it have the potential for self-improvement. Multi-agent model systems are widely used in many fields such as architecture, planning, landscape, and transportation. Although these research applications are still mainly in the academic field, they have not been able to be applied and promoted in the industrial field. However, as more and more designers with programming foundation and thinking join the digital design team, the results of multi-agent systems in the field of digital generative design will be more fruitful. These results can also be packaged into practical tools with commercial value and delivered to the hands of every architectural design practitioner. It is hoped that in the near future, computational generative design will become an essential "weapon" for thousands of architects. Architects can design a variety of rich architectural solutions through a simple interface and the use of complex and efficient programs behind it. So that the creativity of architects can truly be liberated from complicated and meaningless drawings, and the spark of thinking of architects can truly shine in the field of design.

References

1. Biao L (2012) Building generative design—Research on computer generation method of building design based on complex system (in Chinese). Southeast University Press, Nanjing
2. Eckel W (2015) Creating an artificial world with a new kind of cellular automata. arXiv preprint arXiv:1507.05789
3. Doulgerakis A (2007) Genetic programming + unfolding embryology in automated layout planning. UCL, London
4. Scheurer F (2005) A simulation toolbox for self-organisation in architectural design. In: Sariyildiz S, Tuncer B (eds) Innovation in architecture, engineering and computing (AEC), vol 2. Delft University of Technology. Faculty of Architecture, Rotterdam, pp 533–543
5. Biao L, Jingping Q (2009) Research on the generation method of "cellular automata" architectural design—Taking the generation tool of "Cube101" as an example (in Chinese). New Architect 2009(03):103–108
6. Zifeng G (2017) Research on building generation method limited by functional topological relationship (in Chinese). Southeast University