

Applied Mathematics and Nonlinear Sciences

<https://www.sciendo.com>

A generative design method of building layout generated by path

Liu Yong^{1,2}†, Hao Chibiao^{1,3}

¹ Faculty of Innovation and Design, City University of Macau, Macau 999078, China

² Quanzhou College of Technology, Quanzhou, China

³ College of Architecture and Urban Planning, Qingdao University of Technology, Qingdao, China

Submission Info

Communicated by Juan Luis García Guirao

Received July 20th 2021

Accepted December 6th 2021

Available online March 31st 2022

Abstract

The floor plan is a key part of architectural design and has the characteristics of multi-objective evaluation. Traditional methods are often laborious and may lead to re-work due to optimisation, which affects efficiency. Related research uses various algorithms to generate floor plans to improve efficiency. Based on the leading role of architects on floor plans, this paper proposes a generative design method that combines the design process. It takes the space shape as the starting point, simplifies the design process into a mathematical model, and uses Rhino and Grasshopper to complete the algorithm development. Using different algorithm combinations, a large number of new layouts can be explored, and building layouts can be adjusted freely through data optimisation without having to redesign. The design evolution of floor plans of different cases shows that such a generative design method is feasible, reusable, and more efficient than traditional methods.

Keywords: floor plan, path, generative design, generation, algorithm

1 Introduction

Floor plans are an important part of architectural design. Due to the influence of the building function, area, shape and other factors, repeated scrutiny of the layout of the building is a normal state. The optimisation content can be summarised as: (a) adjust the room size and shape to optimise economic and technical indicators or optimise the architectural shape, (b) adjust the number and combination of rooms to improve functions or traffic flow, (c) adjust doors, windows and other components, location and size to optimise daylighting and building facades. There are two main types of layout and optimisation methods: (i) Traditional layout methods. Start by drawing the function bubble diagram, and gradually realise the plane modelling. Due to the direct manipulation of geometric figures, if there are major adjustments such as (a) and (b) of the optimization content, they often

†Corresponding author.

Email address: U17092105112@cityu.mo

need to be redrawn, which affects work efficiency. However, the traditional method is still relatively common. (ii) Automated building layout. Although the degree of automation is different, this type of method is more efficient and can provide multiple solutions at the same time. Since most of these studies are not from a practical point of view, they are not flexible enough in practice, and even inferior to traditional methods. Therefore, architects need a flexible approach. In recent years, generative design has made many useful explorations in design, evolutionary calculation and morphogenesis. Although there are definitional controversies [1], they all emphasise the evolutionary process of design, such as the application of design procedures (DP) to the exploration of column forms [2], the generation of design systems for energy-efficient building solutions [3], using parameters and algorithmic processes to allocate shading equipment [4], etc. However, the exploration of generative design on the architectural layout is rare.

This article reviews some research results of automated building layout, analyses the whole process of traditional methods, and proposes a new method. This method uses algorithm components developed by different parameter models to realise the architectural layout. These parameter models are closely related to the path, which is an important evaluation index of the floor plan. Therefore, the new method is defined as a design method that generates the architectural layout from the path. In short, the contribution of this article is:

- The use of parametric models to study the evolution of building layouts provides new methodological insights.
- Using algorithms developed by parametric models, architects can explore a large number of new layouts in-depth.
- Compared with traditional methods, it can avoid redrawing and improve work efficiency.

2 Related work

Different from the traditional plan space allocation method, Merrell et al. [5] proposed a fully automatic building layout generation method from the perspective of computer games and large-scale scene automatic generation. Specifically, they followed the architect's original bubble chart. The habit of using the data of 120 construction projects to train the Bayesian network to automatically design the floor plan, and to optimise the design through the design value function to achieve the desired goal. Arvin and House [6] proposed a method of applying motion physics to space planning. This method allows designers to create space plans by specifying and modifying graphic design goals instead of directly manipulating the original geometric figures. Ahmed et al. [7] proposed a sketch-based system from the perspective of architects referring to previous experience in similar projects to solve existing problems. Users only need to draw part of a new plan sketch to search for semantically similar plans from the plan library. To generate such a gallery, a new and complete plan analysis system was also proposed. With the emergence of generative adversarial networks (GAN), generative models have made breakthroughs in the field of computer vision [8]. The use of deep convolutions to generate a confrontation network can generate a new bedroom case model [9], and use GAN's pix2pix technology to perform various image conversions, such as converting satellite images into Google maps and converting sketch images into colour pictures [10]. Chaillou [11] used the pix2pix image translation technology based on cGAN neural network in his master's degree thesis to learn the mapping from the input image to output image. The research training generator ArchiGAN is divided into three steps. As part of the tasks in the workflow, the first step is to infer the house occupancy from the shape of the base, the second step is to realise the partitioning and windowing of the house occupancy, and finally the furniture layout is realised. Architects can use the trained ArchiGAN to freely modify or fine-tune the model to achieve human-computer interaction. Nauata et al. [12] proposed a new graph-constrained generative confrontation network, whose generator and discriminator are based on the relational architecture.

The above-mentioned research can be divided into two categories according to different methods. One is

to use machine learning methods to obtain generative models, and the other is to use methods that combine architect experience and knowledge of other disciplines. Because the purpose of research is automation, it is difficult to be widely used directly in design practice. For example, the automatic building layout method based on the Bayesian algorithm transfers the control of each target of the floor plan to the computer. The floor plan generated by the sketch-based system is difficult to be further optimised according to the target requirements, and the data format generated by ArchiGAN is a non-vector format. On the other hand, the above studies all decompose the generated plan into different design goals or solve them step by step, which is consistent with traditional methods.

3 Process analysis of traditional layout method

Although the methods of different architects are different, the process of the traditional layout method can be roughly divided into three stages: (i) draw functional bubble diagrams, study spatial connectivity, (ii) draw axis plan, study spatial geometric relationships, (iii) draw architectural floor plan, drawing walls, adding doors and windows, decorating homes, etc. A typical example is shown in Figure 1.

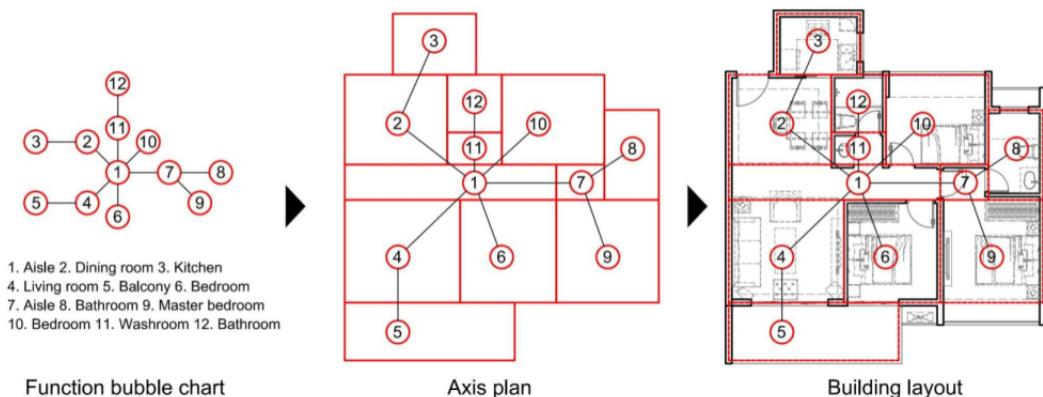


Fig. 1 An apartment adopts the traditional layout method design process

The functional bubble diagram is a sketching system used by architects to consider spatial connectivity. It does not consider the geometric shape, size, orientation and distance of the space, whereas it only considers whether the space is connected or not. The circle is often used to indicate the space, the short line is used to indicate the space connection, and the text inside the circle indicates the space function. It can also be said that the functional bubble chart is a purely spatial connection road map. Axis plan view is a system that uses the wall axis enclosing the space to express the space, and further considers the spatial geometric relationship. The spatial geometric relationship includes the shape, size and orientation of the space. The floor plan of the building is an engineering drawing. Based on the plan view of the axis, details such as wall and door and window components are added. From a mathematical point of view, the entire design process is a process of increasing constraints (Table 1).

Table 1 Beam parts of the bureau method process

Stage	Function bubble chart	Axis plan	Building layout
Constraints	Spatial connectivity	Spatial connectivity Spatial geometry relationship	Spatial connectivity Spatial relationship Graphic geometric relationships

The advantage of the traditional layout method lies in the clear division of labour at each stage. However, due to different constraints, the architect's direct manipulation of graphics, and the irreversibility of the process, traditional layout methods have obvious shortcomings:

- There should be wireless options regardless of the function bubble chart, axis plan or building plan. However, the function bubble chart only chooses one, which is not conducive to the diversity of building layouts (such as changes in the number of spaces and connections).
- The architectural layout is a multi-objective optimisation process, and some important geometric relationship changes (such as architectural modelling, room proportions, area changes) often lead to re-drawing.

4 Generative design method of building layout generated by the path

4.1 Design evolution process

Our method is to use parametric models to develop corresponding algorithms to achieve design evolution instead of directly manipulating graphics, thus avoiding the shortcomings of traditional methods. The design evolution process is still divided into three steps: (i) use the algorithm developed by the path model to generate the axis plan, (ii) use the algorithm developed by the wall model to generate the wall plan, (iii) use the algorithm developed by the door and window model to generate doors and windows. Compared with traditional methods, the path model enables design evolution to support the diversification of spatial layout from the beginning. In addition, we adopted spatial syntax correlation analysis technology [13], and restricted the research scope to orthogonal histograms, and all spaces were divided into convex shapes [14]. However, the real space also has a composite space composed of convex shapes, and the composite space can be generated using the door and window algorithm (Figure 2).

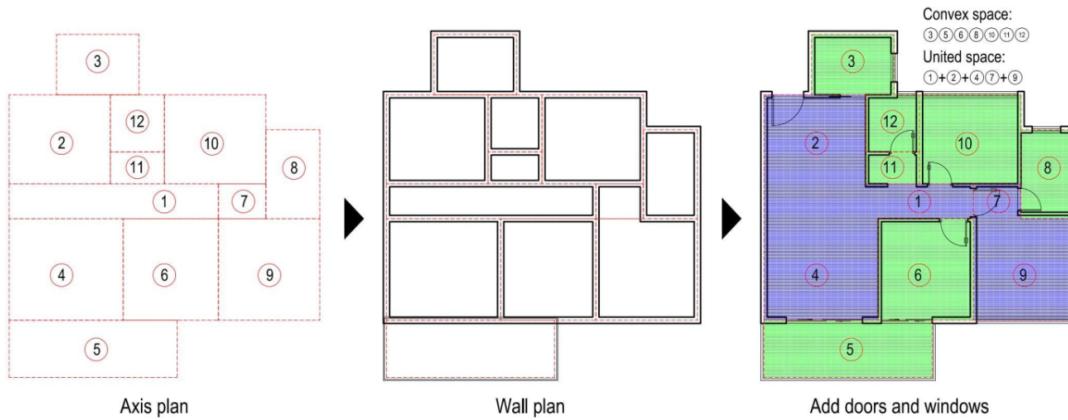


Fig. 2 An apartment uses algorithms developed by parametric models to achieve design evolution

4.2 Parametric model

4.2.1 Path model

The layout problem can be summed up as a combination of space quantity, size and orientation. Among the building types with orthogonal histogram layout (apartment, teaching building, office building, hotel, hospital ward building, etc.), the layout optimisation of apartment plan is more common, and the apartment is still the research object. First select the plane of a high-rise apartment, with the wall axis as the boundary, divide the plan into different convex shapes (rectangles) and number them [14]. Then use points to indicate the location of

the entrance (door or opening), use arrows to indicate the spatial orientation, mark the convex size and sort by orientation. Finally, these points are placed in a rectangular coordinate system to analyse the influence of their position changes on the layout (Figure 3). [13, 14].

Space is divided into five types: upward, downward, leftward, rightward, and any direction. Since the last category can be merged into any of the previous categories, the space is divided into four categories. Specify parameters and variable symbols in terms of orientation, and use a series of numbers instead of a single value to represent variables (Table 2). For example, in the case of Figure 3, the number of upward spaces is 9, and the variables corresponding to the orientation point, length value, and width value each have 9 values. Such regulations can simplify input and realise a variety of combinations of space quantity and size.

Table 2 Four space types and their parameter variable symbols

Spatial orientation	Orientation point (entrance)	Space length	Space width
Upward spaces	{Pu _n }	{Lu _n }	{Wu _n }
Downward spaces	{Pd _n }	{Ld _n }	{Wd _n }
Leftward spaces	{Pl _n }	{Ll _n }	{Wl _n }
Rightward spaces	{Pr _n }	{Lr _n }	{Wr _n }

The orientation points shown in the coordinate diagram in Figure 3 are in a discrete state, showing one of the many possible layouts of the apartment. To add more combinations, a random point P(a, b) is introduced into the coordinate system as a reference (Figure 4), and the horizontal and vertical distances from each point are also defined as a series of numbers. Each point has its own two series of numbers that indicate its coordinates, and four positions correspond to 8 series of numbers. Changing the number of sequence items can adjust the amount of space, and changing the value of the sequence item can adjust the spatial orientation. This optimisation process is essentially the optimisation of the route of the building layout (the azimuth points represent the entrances and connections of different spaces). It only changes the route of travel in space without changing the shape of the route. In comparison, the previous four types of space optimisation process can be regarded as the optimisation of path shape and size. Therefore, we define the parametric model that realises this evolutionary process as the path model, the former is the path positioning model referred to as the positioning model, and the latter is the branch shape optimisation model referred to as the branch model.

Using the positioning model, branch points in different directions can be output and their number and direction can be changed freely. The input parameters are specified as: random point P (a, b), the horizontal and vertical distance between branch point (azimuth point) and random point: ({xun}, [15]), ({xdn}, {ydn}), ({xln}, {yln}), ({xrn}, {yrn}). Please note that in a rectangular coordinate system, with a random point as a reference, the distance between the upper and left sides is positive, and the distance between the lower and left sides is negative.

Using the branch model, you can output rectangles of different orientations and sizes, and can freely change their size. The input parameters are specified as: up, down, left and right branch points Pu, Pd, Pl, Pr, branch length {Lun}, {Ldn}, {Lln}, {Lrn}, branch left width {Wuln}, {Wdln}, {Wlln}, {Wrln} and the right width of the branch {Wurn}, {Wdrn}, {Wrn}, {Wrn}. The left width of the branch and the right width of the branch refer to the distance from the branch point to both sides of the rectangle. In addition, the branch model also includes window positioning coordinate calculation. According to most engineering practices, the positioning point is the midpoint of each side of the rectangle, and no new input parameters are required. Figure 4 shows the relevant variables of the path model.

The coordinates of the branch points in different orientations of the positioning model are calculated as follows:

Pu_n coordinates of the upper branch point:

$$X = a + xu_n \quad (1)$$

$$Y = b + yu_n \quad (2)$$

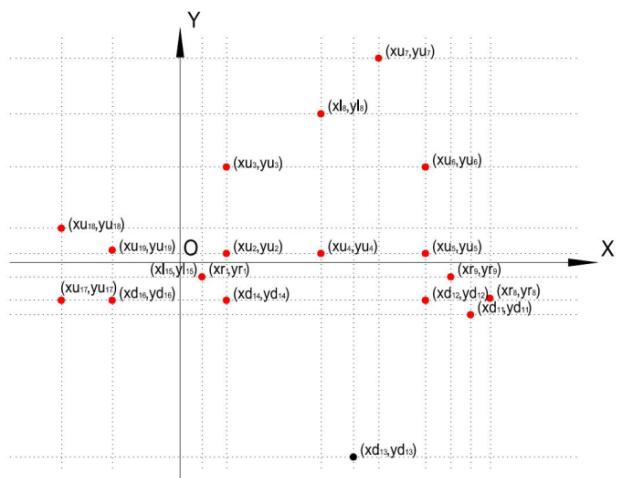
Pd_n coordinates of the downward branch point:

$$X = a + xd_n \quad (3)$$

$$Y = b + yd_n \quad (4)$$



Apartment plan and its space number



Orientation point (entrance) coordinate diagram

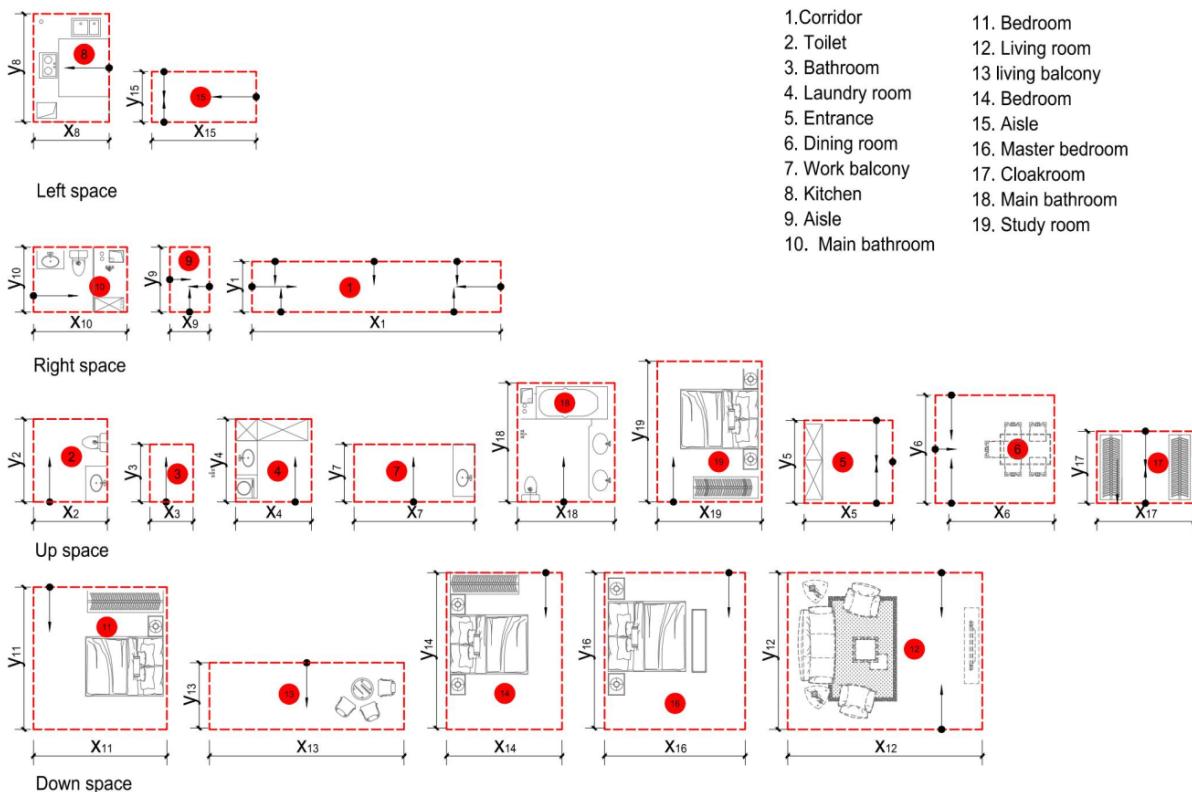


Fig. 3 The plane of an apartment is divided into different convex shapes and classified according to their orientation. Discrete points in a discrete state in the coordinate system show one of the possibilities of apartment layout

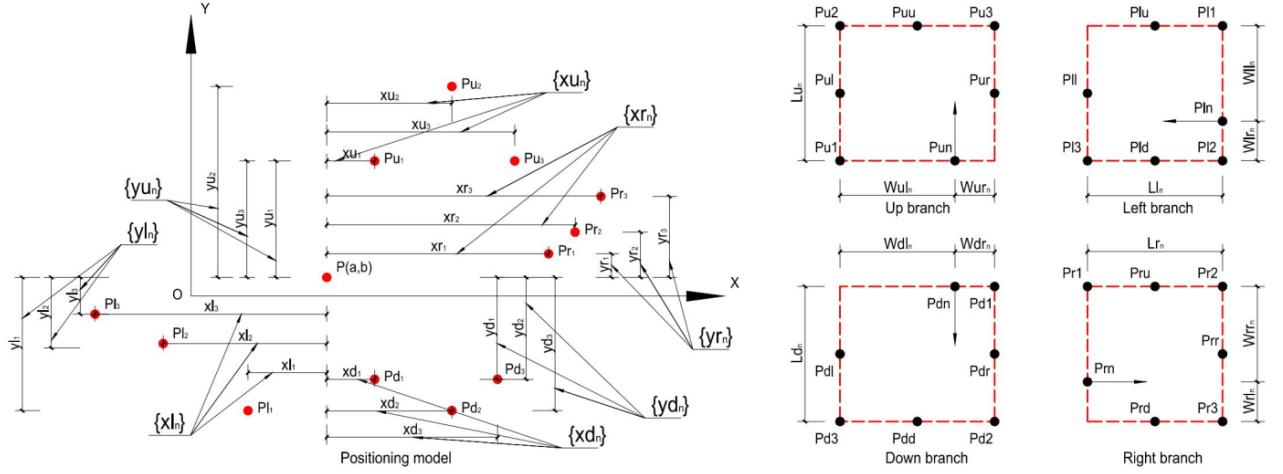


Fig. 4 Related parameters and variables of the path model

Pl_n coordinates of the left branch point:

$$X = a + xl_n \quad (5)$$

$$Y = b + yl_n \quad (6)$$

Pr_n Coordinates of the right branch point:

$$X = a + xr_n \quad (7)$$

$$Y = b + yr_n \quad (8)$$

The rectangle R output by the branch model can be determined by its three corners, and the corner coordinates are calculated as follows:

Coordinates of Ru corner point of upper branch shape:

Pu1:

$$X = a + xu_n - Wul_n \quad (9)$$

$$Y = b + yu_n \quad (2)$$

Pu2:

$$X = a + xu_n - Wul_n \quad (9)$$

$$Y = b + yu_n + Lu_n \quad (10)$$

Pu3:

$$X = a + xu_n + Wur_n \quad (11)$$

$$Y = b + yu_n + Lu_n \quad (10)$$

Coordinates of Rd corner point of downward branch shape:

Pd1:

$$X = a + xd_n + Wdr_n \quad (12)$$

$$Y = b + yd_n \quad (4)$$

Pd2:

$$X = a + xd_n + Wdr_n \quad (12)$$

Pd3:

$$X = a + xd_n - Wdl_n \quad (14)$$

$$Y = b + yd_n - Ld_n \quad (13)$$

Coordinates of Rd corner point of left branch shape:

Pl1:

$$X = a + xl_n \quad (5)$$

$$Y = b + yl_n + Wll_n \quad (15)$$

Pl2:

$$X = a + xl_n \quad (5)$$

$$Y = b + yl_n - Wlrl_n \quad (16)$$

Pl3:

$$X = a + xl_n - Lln_n \quad (17)$$

$$Y = b + yl_n - Wlrl_n \quad (16)$$

Coordinates of Rd corner point of right branch shape:

Pr1:

$$X = a + xr_n \quad (7)$$

$$Y = b + yr_n + Wr rr_n \quad (18)$$

Pr2:

$$X = a + xr_n + Lr_n \quad (19)$$

$$Y = b + yr_n + Wr rr_n \quad (18)$$

Pr3:

$$X = a + xr_n + Lr_n \quad (19)$$

$$Y = b + yr_n - Wr rl_n \quad (20)$$

The window positioning coordinates output by the branch model take the midpoint of each side of the rectangle (except the side where the branch point is located), and the coordinates are calculated as follows:

Positioning coordinates of the upper branch window:

Puu:

$$X = a + xu_n - \left(Wul_n - \frac{Wul_n + Wur_n}{2} \right) \quad (21)$$

$$Y = b + yu_n + Lu_n \quad (10)$$

Pul:

$$X = a + xu_n - Wul_n \quad (9)$$

$$Y = b + yu_n + \frac{Lu_n}{2} \quad (22)$$

Pur:

$$X = a + xu_n + Wur_n \quad (11)$$

$$Y = b + yu_n + \frac{Lu_n}{2} \quad (22)$$

Positioning coordinates of the downward branch window:

Pdd:

$$X = a + xd_n - \left(Wdl_n - \frac{Wdl_n + Wdr_n}{2} \right) \quad (23)$$

$$Y = b + yd_n - Ld_n \quad (13)$$

Pdl:

$$X = a + xd_n - Wdl_n \quad (14)$$

$$Y = b + yd_n - \frac{Ld_n}{2} \quad (24)$$

Pdr:

$$X = a + xd_n + Wdr_n \quad (12)$$

$$Y = b + yd_n - \frac{Ld_n}{2} \quad (24)$$

Positioning coordinates of the left branch window:

Plu:

$$X = a + xl_n - \frac{Ll_n}{2} \quad (25)$$

$$Y = b + yl_n + Wll_n \quad (15)$$

Pll:

$$X = a + xl_n - Ll_n \quad (17)$$

$$Y = b + yl_n + Wll_n - \frac{Wll_n + Wlr_n}{2} \quad (26)$$

Pld:

$$X = a + xl_n - \frac{Ll_n}{2} \quad (25)$$

$$Y = b + yl_n - Wlr_n \quad (16)$$

Positioning coordinates of the right branch window:

Pru:

$$X = a + xr_n + \frac{Lr_n}{2} \quad (27)$$

$$Y = b + yr_n + Wrr_n \quad (18)$$

Prr:

$$X = a + xr_n + Lr_n \quad (19)$$

$$Y = b + yr_n + Wrr_n - \frac{Wrl_n + Wrr_n}{2} \quad (28)$$

Prd:

$$X = a + xr_n + \frac{Lr_n}{2} \quad (27)$$

$$Y = b + yr_n - Wrl_n \quad (20)$$

Table 3 lists the parameter variables and mathematical expression numbers of the path model. The parameter diagram in Figure 5 shows the general relationship between the parameters and the calculation process. The positioning model can be used to generate layout positioning points, and can also freely adjust the number and coordinates of positioning points, to achieve the purpose of freely controlling the layout. With the branch model, the space size can be freely modified.

Table 3 Path model parameters and mathematical expressions

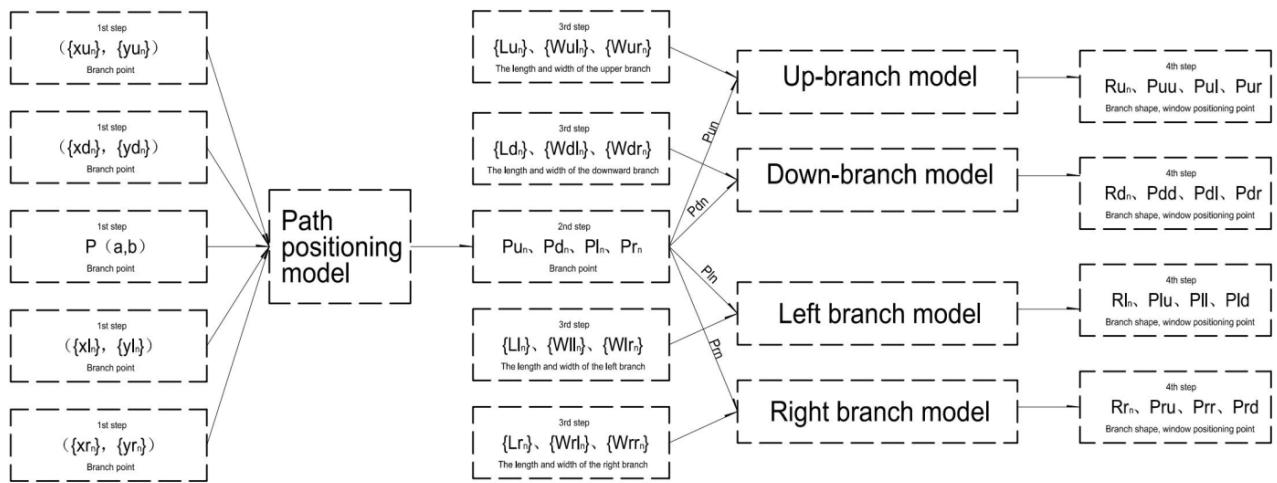
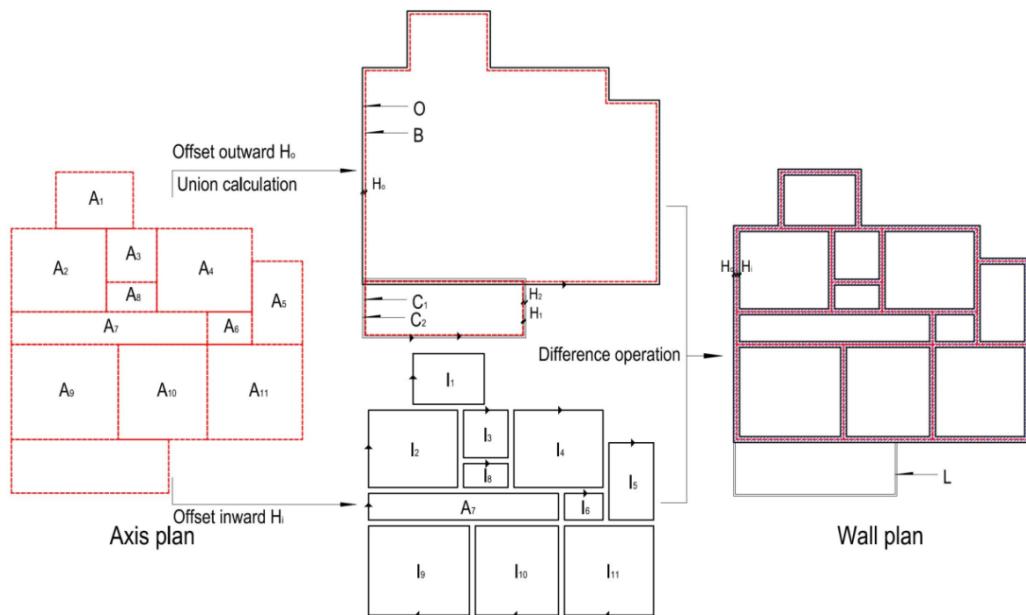
Parametric model category	Input	Output	Mathematical expression
Positioning model	$P(a, b), (\{xu_n\}, [15]), (\{xd_n\}, \{yd_n\}), (\{xl_n\}, \{yl_n\}), (\{xr_n\}, \{yr_n\})$	Branch point (Pu_n, Pd_n, Pl_n, Pr_n)	(1)–(8)
Upper branch	$Pu_n, \{Lu_n\}, \{Wul_n\}, \{Wur_n\}$	Branch shape Run (Pu_1, Pu_2, Pu_3) Window positioning point (Puu, Pul, Pur)	(2), (9)–(11), (21), (22)
Downward branch	$Pd_n, \{Ld_n\}, \{Wdl_n\}, \{Wdr_n\}$	Branch shape Rdn (Pd_1, Pd_2, Pd_3) Window positioning point (Pdd, Pdl, Pdr)	(4), (12)–(14), (23), (24)
Left branch	$Pl_n, \{Ll_n\}, \{Wll_n\}, \{Wlr_n\}$	Branch shape Rln (Pl_1, Pl_2, Pl_3) Window positioning point (Plu, Pll, Pld)	(5), (15)–(17), (25), (26)
Right branch	$Pr_n, \{Lr_n\}, \{Wrln\}, \{Wrrn\}$	Branch shape Rrn (Pr_1, Pr_2, Pr_3) Window positioning point (Pru, Prd, Prr)	(7), (18)–(20), (27), (28)

4.2.2 Wall model

The wall generation can be regarded as the Boolean operation process of subtracting multiple inner rings from one outer ring. The original example is still used to show the entire calculation process. As shown in Figure 6, there are three steps: (i) First, divide the previous path shape into two types. One type can generate walls, which are defined as A1, A2, A3, ..., find their union B. And the obtained new ring B is offset from H_0 to obtain the outer ring O, which is the outer contour of the exterior wall of the building; the other type of path does not generate walls but generates peripheral auxiliary spaces such as balconies, corridors, etc., which is defined as C. Offset them by H_1 and H_2 to obtain two outer rings C1 and C2. Perform the difference calculation between the outer rings C1 and C2 and the outer ring O to obtain the outer contour shape L of components such as balconies and verandahs. (ii) Second, offset all path shapes inward by H_i to obtain multiple inner loops I1, I2, I3, ... as the inner contours of the architectural space. (iii) Finally, perform the difference calculation between the outer ring O and all the inner rings Ii, and the plane shape W of the wall can be obtained. The wall thickness H is

Table 4 Wall model parameters and mathematical expressions

Input	Output	Mathematical expression
Path shape (A_i)	Out ring (outer wall axis B)	$B = A_1 \cup A_2 \cup A_3 \cup \dots$ (29)
Offset distance (H_o, H_i)	Outer wall outline (outer ring O), building space inner outline (inner ring I_i), wall shape W	$W = O - I_1 - I_2 - I_3 - \dots$ (30)
Offset distance (H_o, H_i)	Wall thickness (H)	$H = H_o + H_i$ (31)
Path shape (C), offset distance (H_1, H_2)	Outer ring C_1, C_2 and balcony, verandah outer contour shape L	$L = O - C_1 - C_2$ (32)

**Fig. 5** Path model parameterised data flow and its relationship diagram**Fig. 6** Diagram of the calculation process of wall generation

the sum of the internal and external deviations of the outer ring. Table 4 lists the parameters and mathematical expressions of the wall model.

4.2.3 Door and window model

Generating doors and windows includes three calculation processes, as shown in Figure 7: (i) Use the branch point coordinates and window positioning point coordinates generated by the path model to calculate the corner coordinates of the opening (rectangle) on the wall to obtain the shape of the opening (rectangle). (ii) Perform the difference calculation between the unopened wall and the shape of the opening to generate the wall with the opening. (iii) Use the corner coordinates of the opening, the branch point coordinates and the window positioning point coordinates to generate the plane shape of the door and window. Take the two plane shapes of side-hung doors and sliding windows as examples to show the calculation process. The related symbols are as follows:

Branch point or window positioning coordinates: $P(x, y)$.

The distance between the branch point and the two sides of the opening: x_l, x_r represent the left and right distance of the X-direction opening, y_l, y_r represent the left-right distance of the Y-direction opening.

Distance from outer wall contour line to outer wall axis: H_o

Distance from inner wall edge to wall axis: H_i

Corner of the wall opening: P_{di} means the corner of the door, P_{wi} means the corner of the window.

Opening shape: S_{di} represents the shape of the i-th door, S_{wi} represents the shape of the i-th window.

Wall shape: W represents the shape of the wall without holes, W_o represents the shape of the wall with holes.

The plane shape of doors and windows: D_s represents the shape of a casement door, W_s represents the shape of a sliding window.

According to the branch point or window positioning point coordinate $P(x, y)$, the corner point coordinates of the hole shape can be obtained.

X-direction corner coordinates of the opening shape:

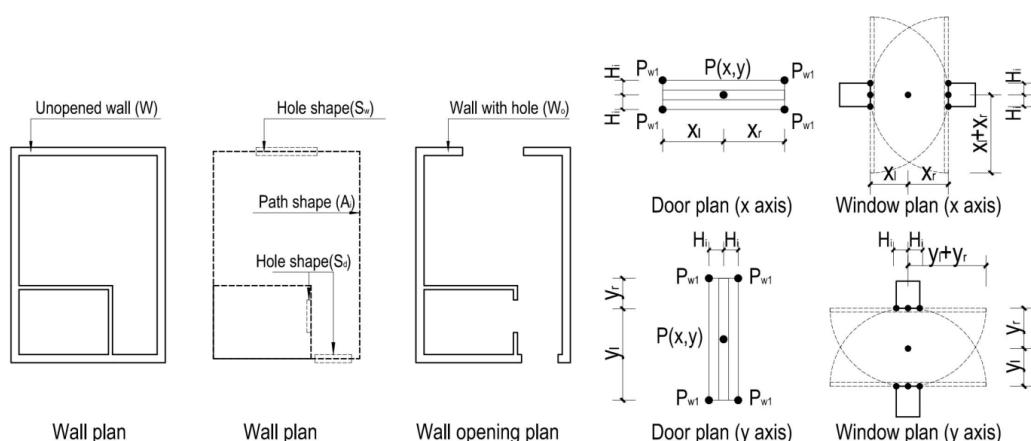


Fig. 7 Calculation diagram generated by doors and windows

$P_{d1}(P_{w1})$ coordinates:

$$X = x - x_l \quad (33)$$

$$Y = y + H_i(H_o) \quad (34)$$

$P_{d2}(P_{w2})$ coordinates:

$$X = x + x_r \quad (35)$$

$$Y = y + H_i(H_o) \quad (34)$$

$P_{d3}(P_{w3})$ coordinates:

$$X = x + x_r \quad (35)$$

$$Y = y - H_i \quad (36)$$

$P_{d4}(P_{w4})$ coordinates:

$$X = x - x_l \quad (33)$$

$$Y = y - H_i \quad (36)$$

X-direction corner coordinates of the opening shape:

$P_{d1}(P_{w1})$ coordinates:

$$X = x - H_i(H_o) \quad (37)$$

$$Y = y + y_r \quad (38)$$

$P_{d2}(P_{w2})$ coordinates:

$$X = x + H_i \quad (39)$$

$$Y = y + y_r \quad (38)$$

$P_{d3}(P_{w3})$ coordinates:

$$X = x + H_i \quad (39)$$

$$Y = y - y_l \quad (40)$$

$P_{d4}(P_{w4})$ coordinates:

$$X = x - H_i(H_o) \quad (37)$$

$$Y = y - y_l \quad (40)$$

According to the above coordinate points, the shape of the opening S_{di} , S_{wi} and the shape W of the wall without opening before are obtained. The mathematical expression of the wall W_o with the opening is:

$$W_o = W - \sum_{i=1}^n S_{di} - \sum_{i=1}^n S_{wi} \quad (41)$$

Swing doors have four combinations of opening modes: inside, outside, left, and right (Figure 7). The opening direction is represented by a quarter arc, the arc is centred at the intersection of the edge of the hole and the axis of the wall, and the door leaf is represented by a straight line segment. The shape of the swing door D_s

includes two parts: a straight line segment and a quarter arc. The mathematical expression of arc radius R and straight line length L is:

$$\text{X direction opening} \quad R(L) = x_l + x_r \quad (42)$$

$$\text{Y direction opening} \quad R(L) = y_l + y_r \quad (43)$$

According to the opening direction of the door, the straight line segment has four combinations of starting and ending points: upper left, upper right, lower left and lower right. If you consider the direction of the opening, there will be eight combinations. The mathematical expression of the starting and ending point coordinates of the segment is:

Starting point coordinates:

$$X = x - x_l \quad (33)$$

$$Y = y \quad (44)$$

End point coordinates:

$$X = x - x_l \quad (33)$$

$$Y = x_l + x_r \quad (45)$$

The plane shape Ws of the sliding window is composed of the hole shape Swi and the window sash shape. The shape of the window sash is the same rectangle, and its thickness is set to d, and the coordinates of the four corners are also obtained from the branch point or window positioning point coordinates P(x, y). Still taking the X-direction opening as an example to show the calculation process, the corner coordinates of the window sash are:

Upper left corner:

$$X = x - x_l \quad (33)$$

$$Y = y + d/2 \quad (46)$$

Upper right corner:

$$X = x + x_r \quad (35)$$

$$Y = y + d/2 \quad (46)$$

Bottom right point

$$X = x + x_r \quad (35)$$

$$Y = y - d/2 \quad (47)$$

Bottom left point

$$X = x - x_l \quad (33)$$

$$Y = y - d/2 \quad (47)$$

Table 5 summarises the parameters and mathematical expressions of the door and window model. Combining the path model and the wall model, and by using the door and window model, we can generate a composite space by freely adjusting the position and size of the doors and windows to optimise various goals such as traffic flow, daylighting, building facades, etc.

Table 5 Model parameters and mathematical expressions of doors and windows

Model category	Input	Output	Mathematical expression
Swing door	$P(x, y), x_l, x_r, y_l, y_r, H_i$	Holtal $S_{di}(P_{d1} P_{d2} P_{d3} P_{d4})$ Form of Ds (Direct-section + arc)	(33)–(40) (33), (42)–(45)
Sliding window	$P(x, y), x_l, x_r, y_l, y_r, H_i, H_o$	Holtal $S_{wi}(P_{w1} P_{w2} P_{w3} P_{w4})$ Window Form $W_s(S_{wi} + \text{Rectangle})$	(33)–(40) (33), (35), (46), (47)
Wall openings	S_{di}, S_{wi}, W	Hole W_o	(41)

4.3 Algorithm development

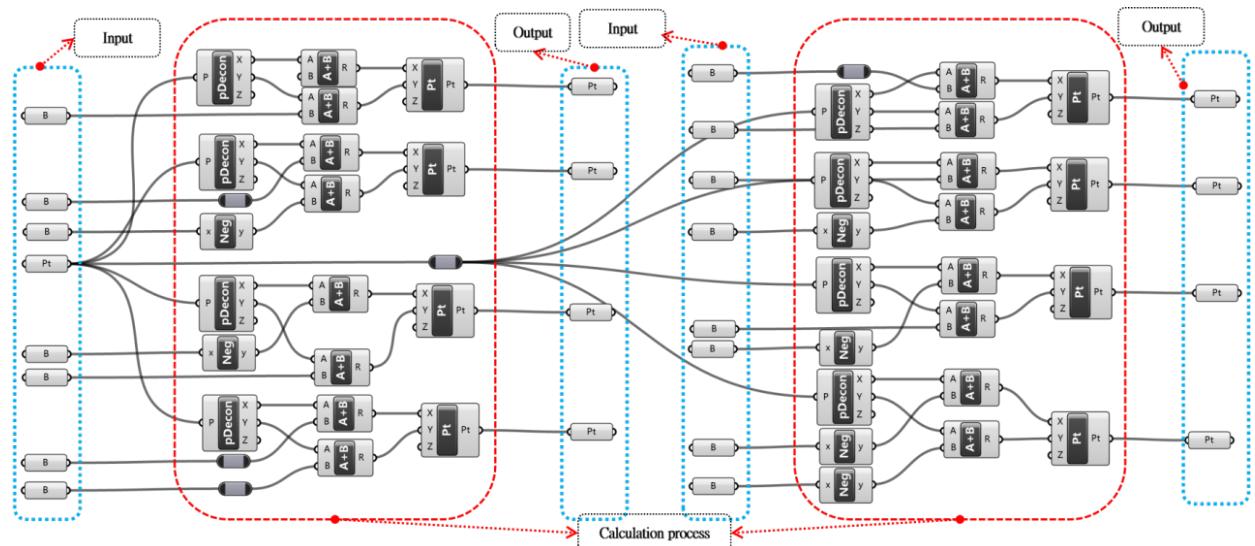
4.3.1 Tools and development ideas

To integrate the parametric model into the algorithm, we used Rhinoceros and Grasshopper. Grasshopper is a visual programming language that runs in the Rhino environment. It can write parametric models into algorithms. Rhino is widely used by architects and planners to help test the applicability of algorithms. The algorithm development idea emphasises flexibility while taking into account efficiency. The input and output of the algorithm are further summarised, and some calculation processes are split and merged.

4.3.2 Algorithm for generating path

The branch point data generated by the positioning model has two flow directions: one is output to the branch model to generate branch shape data; the other is output to the door and window model to generate door, window and opening shape data. The further output of the branch shape data also has two flow directions: one is used as the wall axis to generate the wall, and the other is used as the auxiliary space axis to generate the outline of the auxiliary space such as balconies and corridors. Therefore, the positioning algorithm for generating branch points should have two data streams, namely wall generation and contour generation, on input and output. Figure 8 shows the input, output and calculation process of the positioning algorithm. Consistent with the branch model, algorithms for generating branch shapes are also divided into four types: up, down, left and right. Figure 9 shows the input, output and calculation process of the branch algorithm as an example.

Use Grasshopper to encapsulate the input, output and calculation process, making it a general algorithm tool. Combine different algorithms to form a continuous data stream, which can realise multiple layouts and

**Fig. 8** Diagram of the positioning algorithm for generating branch points

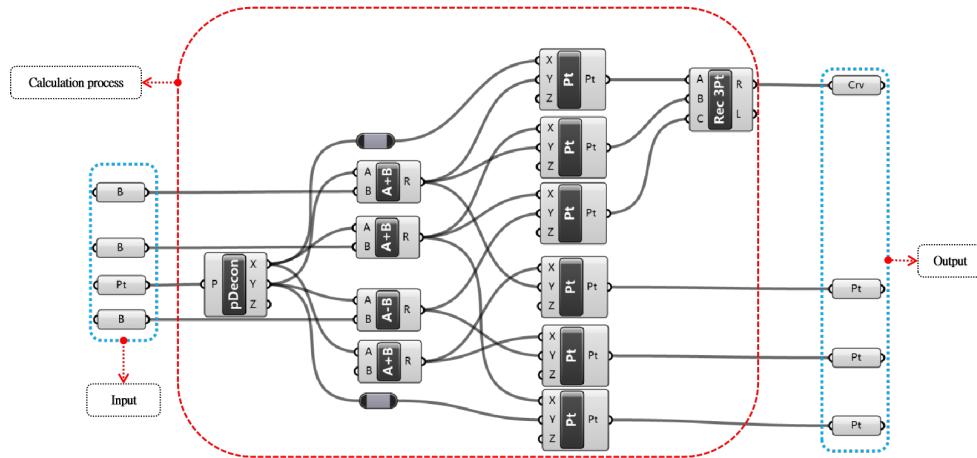


Fig. 9 Diagram of branch algorithm for generating branch shape and window positioning point

can be adjusted freely. Figure 10 shows the potential of using algorithms to generate and adjust layouts. Using only one algorithm and randomly inputting a series of branch points, multiple layouts can be generated. Using the combination of positioning algorithm and branching algorithm, the spatial orientation can be adjusted freely. Table 6 analyses the parameters and usage of the path algorithm in Figure 10.

4.3.3 Generated algorithm

Based on the continuity of the calculation process and the flexibility of the algorithm tools, the calculation process of generating doors and windows is divided into two parts: the Boolean operation of opening the wall and the calculation of generating the shape of the opening. The former is incorporated into the calculation process of generating the wall. The algorithm for generating the wall includes three parts of calculation: generating the wall section W without opening, generating the wall section Wo with opening, and generating the auxiliary space contour L. To observe the relevant area indicators at any time, the calculation process of the relevant technical indicators is also included as part of the algorithm. The input, output and calculation processes are shown in Figure 11.

Table 6 Analysis of algorithm usage for generating paths

Algorithm application	Input	Output analysis
Up-branch algorithm, used alone	Randomly input 6 branch points, a total of 5 times, the following input remains unchanged: Lu1=1, Lu2=1.5, Lu3=1, Lu4=1.5, Lu5=1.5, Lu6=1.7, Wu ln=0.6, Wur1=2, Wur2=3, Wur3=3.5, Wur4=2, Wur5=2.5, Wur6=3.5	Figure a: Randomly changing the position of the branch point can generate a new layout
Combination of positioning algorithm and upper branch algorithm	Change the item value and number of {xun} and [15], the following parameters remain unchanged: Lu1= 3, Lu2= 3, Lu3= 2, Lu4= 2, Wuln=0.6, Wurn=2	Figure b: Using positioning algorithms, you can freely adjust the new layout according to the design goals
Up-branch algorithm, used alone	Change the item values of {Lun}, {Wuln}, {Wurn}, and randomly input points P1, P2, P3, P4 and keep them unchanged	Figure c: Changing the value of a series item can generate a new layout

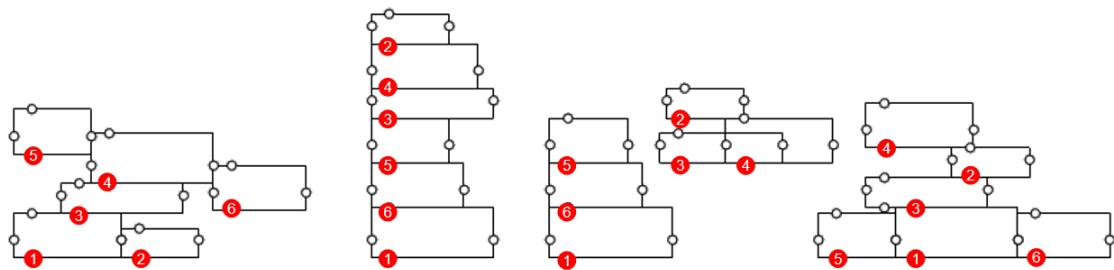


Figure a

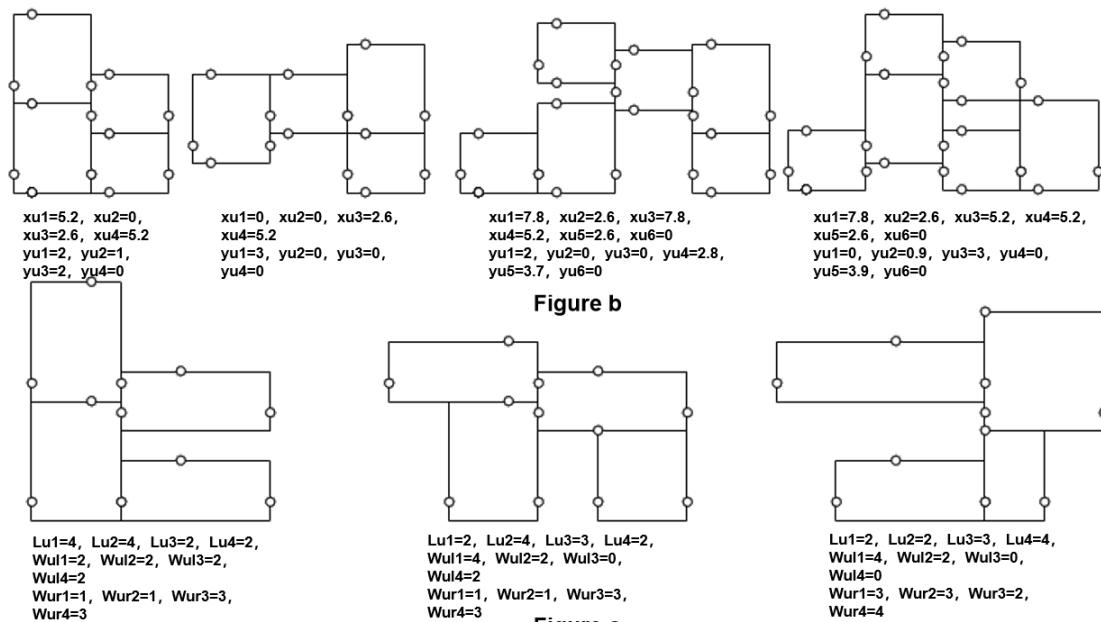


Figure b

$Lu1=4, Lu2=4, Lu3=2, Lu4=2,$
 $Wu1=2, Wu2=2, Wu3=2,$
 $Wu4=2$
 $Wur1=1, Wur2=1, Wur3=3,$
 $Wur4=3$

Figure c

$Lu1=2, Lu2=2, Lu3=3, Lu4=4,$
 $Wu1=4, Wu2=2, Wu3=0,$
 $Wu4=0$
 $Wur1=3, Wur2=3, Wur3=2,$
 $Wur4=4$

Fig. 10 Path algorithm is being used to generate multiple layouts

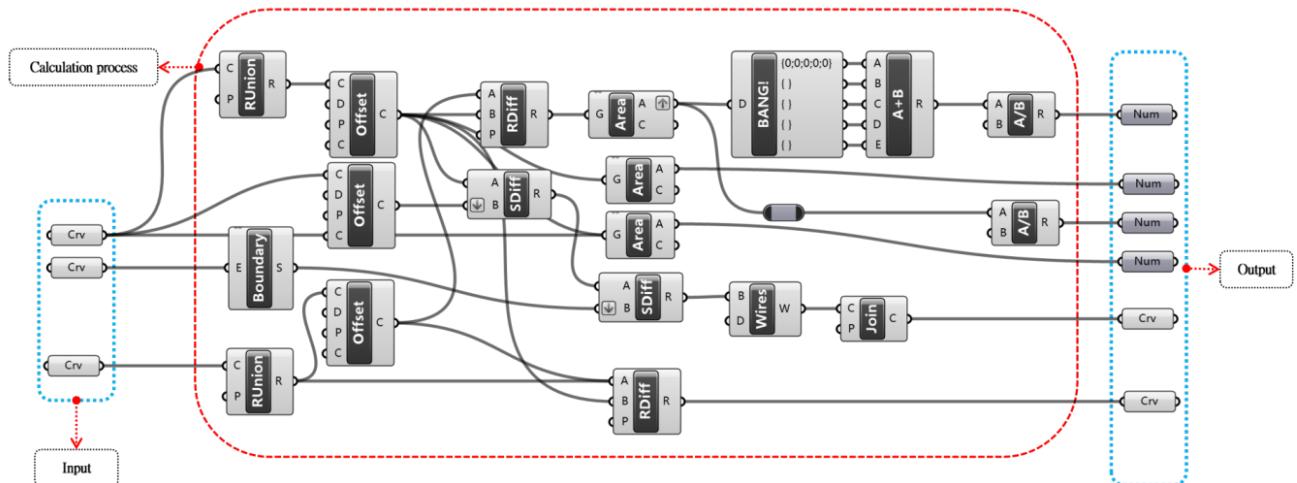


Fig. 11 Algorithm for generating walls

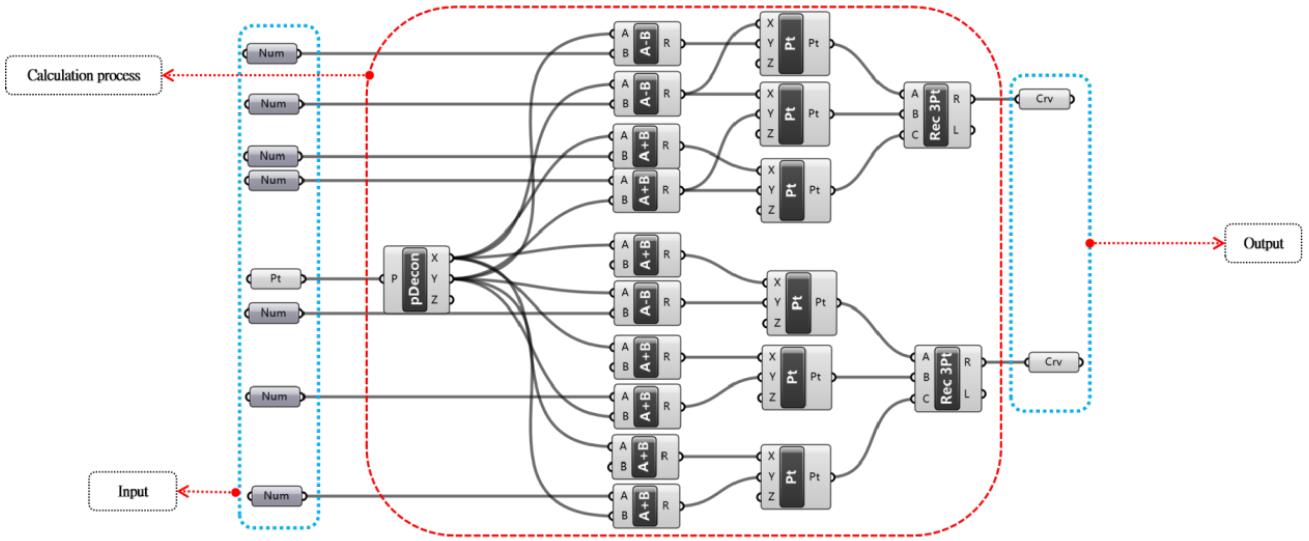


Fig. 12 Algorithm for generating window (upper and downward position)

4.3.4 Algorithm for generating doors and windows

According to different directions, the algorithms for generating doors and windows are also summarised into two categories: generating upper and lower positions and generating left and right directions. To improve efficiency, the algorithm input setting defines all the input parameters such as the length value of the door and window, the width value and the number of connected points as a series of numbers, which can generate doors and windows of different sizes at one time. In addition, compared with sliding windows, side-hung doors have four possible opening directions: inward to left, inward to right, outward to left, and outward to right. The output item also includes the option of opening direction, which can be used when using the algorithm. Freely choose the opening method. Figure 12 shows the input, output and calculation process of generating upper and lower bit windows.

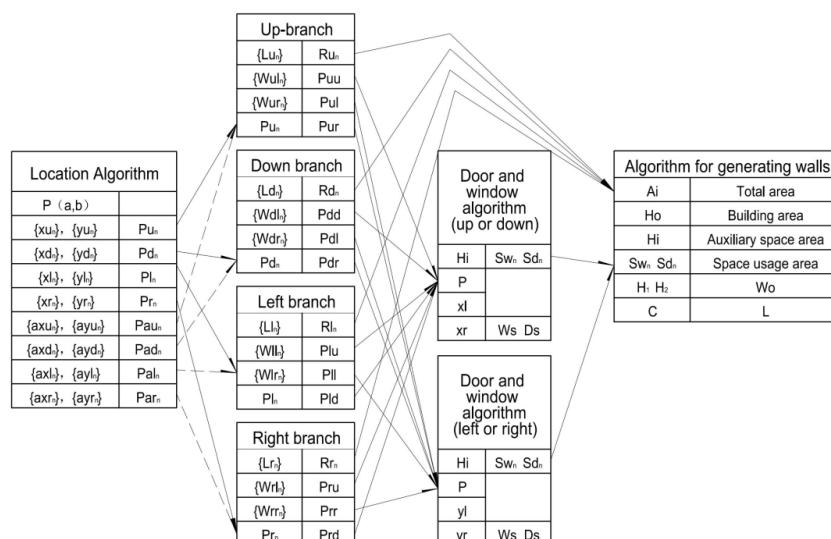


Fig. 13 Algorithm data flow diagram

4.3.5 Algorithm and design evolution

The use of algorithms to achieve layout evolution can be summarised as data processing. Inputting data such as random points and relative distances at the input of the positioning algorithm can output branch points in different orientations; inputting the branch points, path length and width data at the input of the branching algorithm can output path shapes and window positioning points in different orientations. The door and window algorithm input terminal inputs data such as branch points, window positioning points, wall thickness, and door and window width. The door and window algorithm can output the shape of the opening and the shape of the door and window. The data of the path branch shape, the shape of the opening and the auxiliary space axis can be input to the wall algorithm and output The contours of the walls and ancillary spaces. Freely combine different algorithms and adjust their inputs to obtain evolutionary results at any time. Figure 13 lists the input, output, and data flow relationships of all algorithms.

5 Case verification

5.1 Case

From the point of view of adjustment, take the graphic design process of a high-rise apartment as an example to verify the feasibility of the algorithm. From the point of view of versatility, use a hotel layout verification algorithm to process the versatility of floor plans of different building types, which can be summarised as: exploring the potential of new layout, efficiency and algorithm reusability.

5.2 Explore the layout

The floor plan shown in Figure 14 is the result of four rounds of layout optimisation for a set of houses in the case. The design and adjustment contents mainly include: (i) room size optimisation, (ii) room quantity change, (iii) room orientation optimisation, (iv) optimisation of apartment types and room area indicators. Figure 15 lists the data input and output of the generation process with the first optimisation result as an example. By changing the coordinate data at the input end of the positioning algorithm and the input end data of the upper branch algorithm, the 2–4 layout optimisation results can be obtained at any time, showing the great potential of exploring new layouts. During the optimisation process, the use of wall algorithms can provide real-time feedback on indicators such as building area, room usage area, balcony area, etc., without the need for additional calculation and feedback to avoid design repetition.

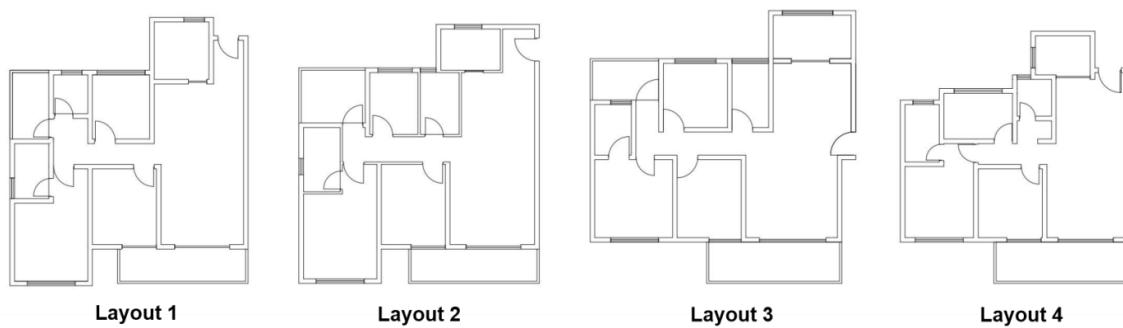


Fig. 14 Use algorithms to optimise some results of the layout of a high-rise apartment

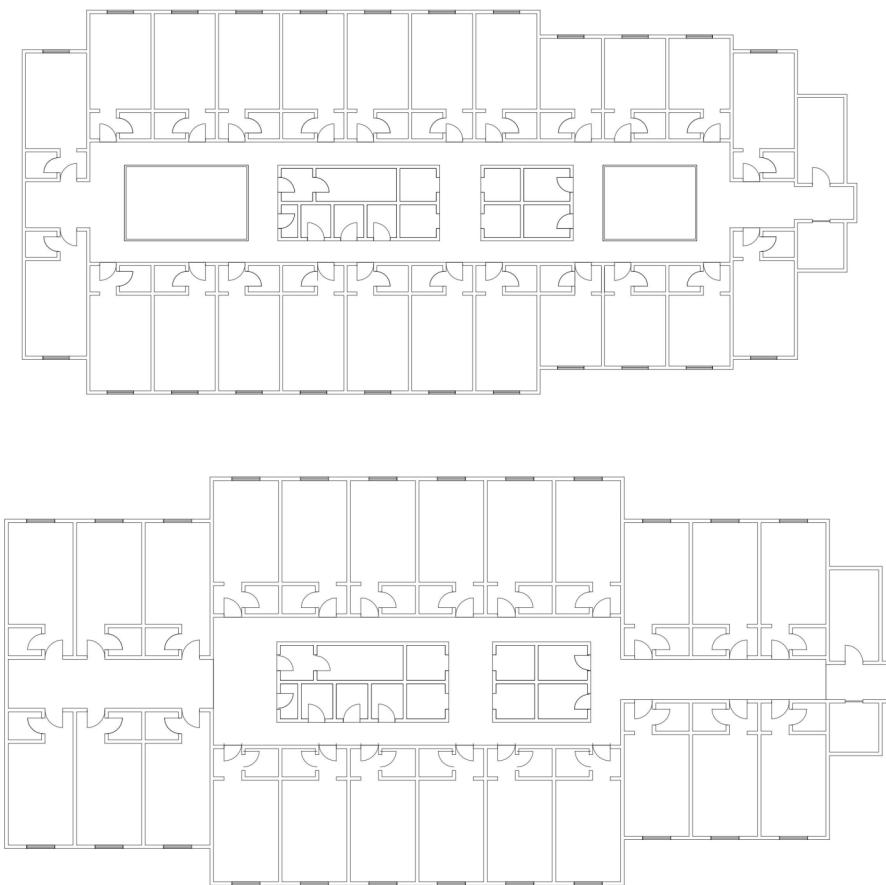


Fig. 15 Use algorithms to optimise the layout of the standard floor of a hotel

5.3 Efficiency

Compared with traditional methods, it is more efficient to use algorithms to explore and optimise the layout of the building. These are explained as follows: (i) A large number of new layouts can be explored from the beginning, and the evolutionary results of each stage are fed back at the same time, (ii) Avoid direct adjustment of graphics, instead optimise the building layout through data driving, (iii) Real-time feedback of technical indicators such as economic indicators based on the results of layout adjustments can help avoid making repeated designs.

5.4 Reusability

Compared with apartments, corridor-style buildings have a large amount of space but few types of space, and it is more advantageous to use algorithms to complete such design tasks. Figure 16 shows the results of two rounds of optimisation of the hotel floor plan. The optimisation content includes: room size, number and type of rooms, area indicators, and adjustments to architectural modelling (Figure 16). The evolution of the hotel layout shows the reusability of the algorithms used in the generative design method.

6 Discussion

From a computational point of view, the architectural layout can be regarded as a search problem for obtaining a large number of possible solutions in the balance of many goals. Using parametric models such as the path positioning model and branch model to describe the plane layout provides designers with new ideas in methods.

Using different combinations of algorithms to find solutions in a wide range can sometimes be very enlightening for designers.

The purpose of the algorithm is to provide a framework for the operation of the transformed geometric figures during the design process. The advantages of using these algorithms in the design include:

1. The graphics can be changed without deleting the redrawing, which provides flexibility for design exploration.
2. Improve the reusability of the solution through packaging.
3. When changes affect other targets, real-time feedback is got.

Compared with the previous research literature, based on the complexity of the architectural layout, the emphasis on the architect's leading role in the design goal has more practical application value for the architect.

Our method also has its limitations. First, it is limited to the architectural layout of the orthogonal histogram type. In addition, this method also has room for improvement in terms of efficiency. For example, the way doors and windows are generated can be more automated.

7 Conclusion

It is a feasible method to use a series of numbers to establish a parametric model to describe the evolution of the architectural space layout. Integrating the parametric model into the algorithm, and inputting or adjusting data at the input of the algorithm can quickly realise planar modelling and optimisation without redesigning. It is more inspiring for architects to freely combine different algorithms and explore a large number of new layouts on a larger scale. The use of algorithms to complete the design evolution of high-rise apartments and hotels shows that compared with traditional methods, architects can explore a large number of new layouts in-depth, with higher efficiency, and generative design algorithms are reusable.

The generative design method has two main limitations: First, it is limited to the architectural layout of the orthogonal histogram type. Under the existing framework, optimising the branch algorithm is a feasible option. Second, in terms of improving algorithm efficiency, the ways and means to automate the generation of doors and windows require further research.

Acknowledgements.

The authors acknowledge the Natural Science Foundation of Shandong Province (Research on Generating Techniques of Curly Grass Patterns in Non-standard Buildings Project Number: ZR2018PEE023).

References

- [1] Caetano I, Santos L & Leitão A (2020) Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research* 9:287–300.
- [2] Barrios Hernandez CR (2006) Thinking parametric design: introducing parametric Gaudi. *Design Studies* 27:309–324.
- [3] Caldas L (2008) Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics* 22:59–70.
- [4] Queiroz N, Dantas N, Nome C & Vaz C. *Designing a Building envelope using parametric and algorithmic processes*. (2015).
- [5] Merrell P, Schkufza E & Koltun V. in *ACM SIGGRAPH Asia 2010 papers on – SIGGRAPH ASIA '10* (2010).

- [6] Arvin SA, House DH (2002) Modeling architectural design objectives in physically based space planning. *Automation in Construction*.
- [7] Ahmed S, Weber M, Liwicki M, Langenhan C, Dengel A & Petzold F (2014) Automatic analysis and sketch-based retrieval of architectural floor plans. *Pattern Recognition Letters* 35:91-100.
- [8] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A & Bengio YJaPA (2014) Generative adversarial networks.
- [9] Gidaris S, Singh P & Komodakis NJaPA (2018) Unsupervised representation learning by predicting image rotations.
- [10] Isola P, Zhu J-Y, Zhou T & Efros AA. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125-1134.
- [11] Chaillou SJHU (2019) AI+ Architecture: Towards a New Approach.
- [12] Nauata N, Chang K-H, Cheng C-Y, Mori G & Furukawa Y. in *Computer Vision – ECCV 2020*. (eds Andrea Vedaldi, Horst Bischof, Thomas Brox, & Jan-Michael Frahm) 162-177 (Springer International Publishing).
- [13] Bafna SJE & Behavior (2003) Space syntax: A brief introduction to its logic and analytical techniques. 35:17-29.
- [14] Yu Z & Jianguo W (2004) On "Space Syntax" Again. *The Architect* 3:33-44.
- [15] Fischer G, Giaccardi E, Ye Y, Sutcliffe AG & Mehandjiev NJCOTA (2004) Meta-design: a manifesto for end-user development. 47:33-37.