B. Dave, A. I. Li, N. Gu, H.-J. Park (eds.), *New Frontiers: Proceedings of the 15th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2010*, 165–174. ©2010, Association for Research in Computer-Aided Architectural Research in Asia (CAADRIA), Hong Kong

AUTOMATIC PLANNING OF RESIDENTIAL QUARTER UNDER INSOLATION CONDITION BASED ON MULTI-AGENT SIMULATION

GUOHUA JI and HUIJIE LIU
Nanjing University, Nanjing, P.R.China
jgh3020@yahoo.com.cn

Abstract. Based on multi-agent simulation principle, this study establishes an automatic layout model for planning residential buildings under the constraint of insolation condition, programmed with NetLogo. According to the residential planning regulations, our model respectively deals with two kinds of constraint: sunshine spacing and sunshine duration.

Keywords. Residential planning; multi-agent simulation; NetLogo; sunshine spacing; sunshine duration.

1. Introduction

Residential quarter planning usually confronts many interrelated problems such as apartment type, floor area ratio, green space, sunlight, as well as form and other design targets, in which insolation standard is a basic and usually a critical constraint. In conventional planning process, the designer usually makes one or several preliminary layouts first based on his understanding of the planning requirements and his premeditated targets, then checks whether the requirements are satisfied. If not, the layout(s) will be adjusted through changing the state of planned objects and checked again, until solutions meet all the requirements and targets. Otherwise, the designer should conceive new preliminary layout(s) and run the process again. Such repeated trial-and-error process is suitable for computational program to do, which can evolve at least the preliminary layouts that meet the requirements or produce satisfactory solutions for the designer to refer.

The trial-and-error process of design can be alternatively regarded as a self adaption process by design elements through the rules of behavior of the individual elements as well as the interactions between them. Although the objects in residential quarter planning, such as housing buildings, road system, green spaces, and public buildings, are not active subjects, in planning process the designer's operations such as changing dimensions, rotating, and moving can be transferred to behaviors of the planning objects, so that the process of planning can be seen as the process in which planning objects change their attributes to search the solutions of planning targets. In this way we can apply the principle of multi-agent simulation to develop a computational device to automatically generate residential quarter plans.

The application of multi-agent systems in architecture and urban planning area is quite new, mainly in two directions: (1) simulation of human behavior in built environment, and (2) simulation of the evolution of cities or habitats. The typical examples include the study on the phenomena of pedestrians by Center for Advanced Spatial Analysis at University College, London (Castle and Crooks, 1994; Batty, 2003), the simulated city model developed by Northwestern University and Illinois Institute of Technology (Lechner et al., 2004; Lechner et al., 2006), and so on. Recently a number of studies applying agent simulation in computational architectural and urban design have been published (Krause, 1997; Testa et al., 2000; Canepara et al., 2005), which discuss various problems of building agent simulations for design.

Li Biao's work (2007, 2008) demonstrated the use of multi-agent simulation in dealing with special concrete problems in design and planning. His model of "HighFar" (Li et al., 2008) developed an automatic mechanism for searching high density residential quarter layouts. But the model was still rough: the buildings were abstracted to circles and various situations of sun shadow were not fully considered.

Our study is a sort of amelioration of the model. This paper introduces our efforts in using the principle of multi-agent simulation to develop a system of automatic planning for residential quarter. We take the problem of sunshine as case study, currently using NetLogo as experimental platform.

2. Definition of housing plan and site plan

NetLogo is a multi-agent programming language and integrated modeling environment developed by Northwestern University's Center for Connected Learning and Computer-Based Modeling. In NetLogo, there are three types of agents: turtles, patches, and the observer. Turtles are agents that move around in the "world". The "world" is two dimensional and is divided up into a grid of patches. Each patch is a square piece of "ground" over which turtles can move. The observer can control turtles and patches through codes.

Because geometrical objects are not well supported in the “world” of NetLogo, we have to find alternative ways to describe them. We abstract the housing building plan to a rectangle, setting 5 synchronous turtles respectively located on the central point and the four corner points to define the plan. The central turtle is the positioning base, and the position of the corner turtles are calculated with the width and the depth of the building plan. The lines connecting corner points are figured by links, which are special turtles in line shape, to indicate the outline of the building (figure 1).

The housing buildings must be planned in a certain confine of a site. We define the site in NetLogo using the attributes of patches. In the planning process, a housing plan defined as agents can distinguish whether it is inside the site through the attribute of the patches underneath.

3. Automatic planning under the constraint of sunshine spacing

3.1. THE CONTROLLING METHOD FOR THE CONSTRAINT OF SUNSHINE SPACING

The minimum row distance between housing buildings can be calculated through the coefficient of sunshine spacing, and the minimum fireproofing spacing between buildings which is basically a constant. In this case, we can assume that there is a forbidden area around each housing building that other buildings should not intersect. As Figure 2 shows, this area is defined by the width of the building, the height of the building and the coefficient of sunshine spacing, as well as the fireproofing spacing.

When the shapes of the buildings are the same and the buildings are facing same direction, the relationship between two buildings has four situations shown in Figure 3. The distances between two buildings’ east-west walls and south-north walls can be calculated through the coordinates of the central turtles with the width and the depth of the buildings as well as the orientation angle. Based on the distances, the system judges whether there are other buildings in the forbidden area of a building. If yes, the building will move toenlarge the distance to the other buildings. This process is programmed in the procedure Position-Searching, based on the rules of OPEN-MIN, OPEN-

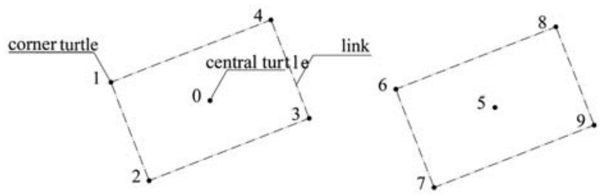

Figure 1. The abstract plan defined through turtles.

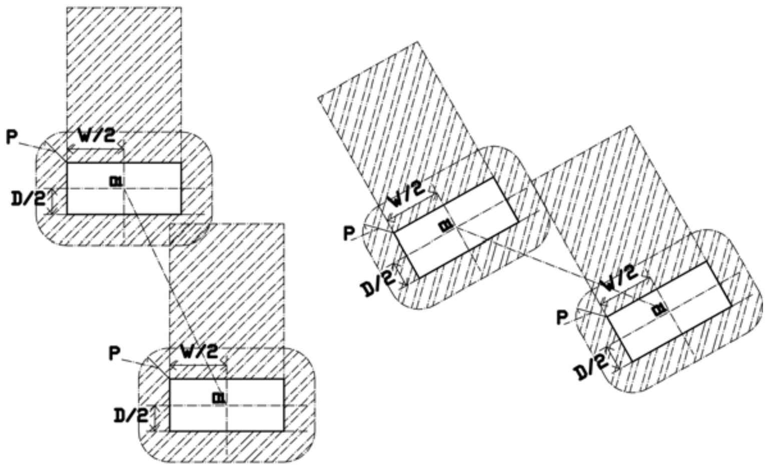

Figure 2. The forbidden area of a building.

MIN-MAX, and RANDOM-AWAY of the Scattering Model published on NetLogo home page (Wilensky, 1998). Two searching methods have been tested: (1) finding out the nearest building and moving in reverse to the direction connecting the central turtles of the two buildings, and (2) calculating the average coordinates of all the buildings in the forbidden area and moving in reverse to the direction connecting the central turtle to the point of the average coordinates of those buildings' central turtles. The former is faster in each step calculation, while the latter is better in mastering overall situation and its total

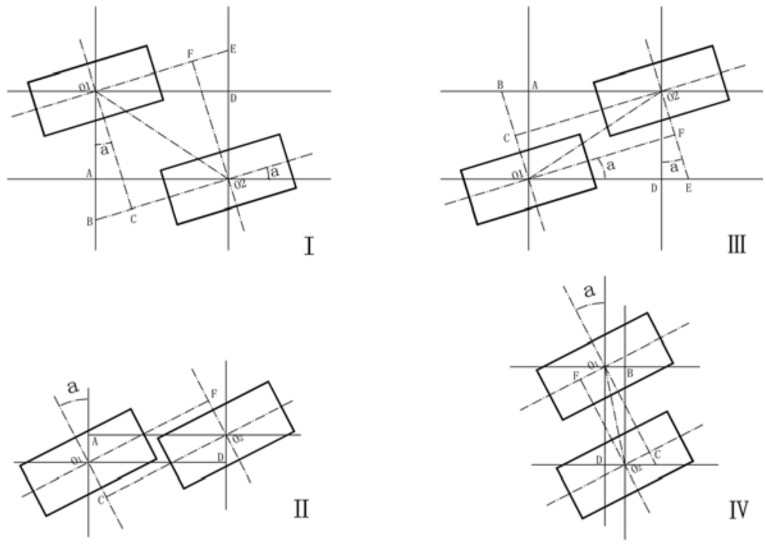

Figure 3. The relations between two buildings.

times of calculation is less. If the building moves out of the site, a procedure will be called to put it randomly inside.

3.2. PLANNING OF DIFFERENT-SHAPED BUILDINGS

A housing quarter normally has more than one type of housing buildings. In the case of the planning of multi-floor housing buildings, the forbidden area of each type of building varies according to its plan size and height. Our system has considered the mixture of two types of buildings, dividing the sunshine obstruction into four situations: Type A building(s) shading a Type A building, Type A building(s) shading a Type B building, Type B building(s) shading a Type A building, and Type B building(s) shading a Type B building. Accordingly we store the Type A building(s) in the forbidden area of a Type A building in the set A-A, and use the sets A-B, B-A and B-B for the other situations respectively. Figure 4 shows the flow of position-searching.

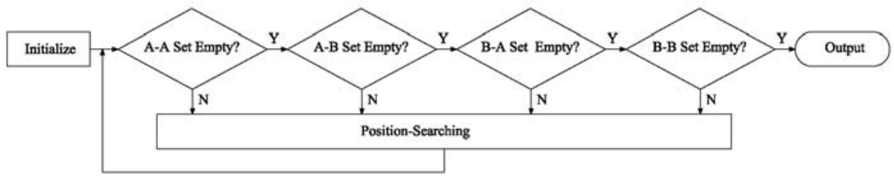

Figure 4. Flow of position-searching for the mixture of two types of buildings.

3.3. PARAMETERS, INTERFACE AND RESULTS

The parameters of the system of automatic planning under the constraint of sunshine spacing include the number of housing buildings, the width, depth, number of floors, and height between floors of the buildings, as well as the minimum fireproofing spacing, the coefficient of sunshine spacing, the orientation angle, the pace that the buildings move, and the initial distributing range of the buildings. Figure 5 shows the interface of the system. The user firstly clicks the “Edit” button to input the coordinates of site border, uses the sliders to adjust the parameters, clicks “Setup” to initialise, and then clicks the “planning” button to start the process to find out the layout that satisfy the constraint of sunshine spacing. The user can activate the “click nodes” button and then drag housing buildings to new positions to intervene in the process, or click the “planning” button to pause the process at any time. Figure 6 shows two results of the same parameters shown in figure 5: the site is 200 by 100 meters, in which there are 20 housing buildings of 55 by 9 meters and 25 buildings of 38 by 13 meters; and the buildings are facing 10 degrees east of south.

4. Automatic planning under the constraint of sunshine duration

The above system can be only applied in the planning of multi-floor housing buildings, while the automatic planning system based on sunshine duration is more universal because the Code for Planning and Design on Urban Residential Areas of P.R. China expressly stipulates the sunshine duration of housing on the day of Mid Winter (approx. Dec 22) or the day of Great Cold (approx. Jan 20), and the coefficient of sunshine spacing is a substitute of sunshine duration specially for the planning of multi-floor housing buildings.

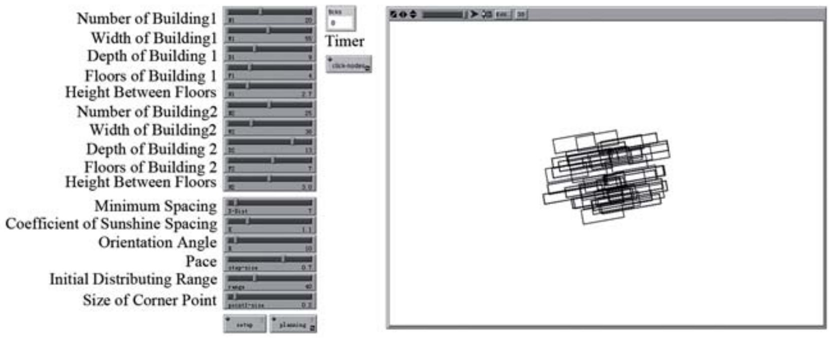

Figure 5. The interface of the system of automatic planning under the constraint of sunshine spacing.

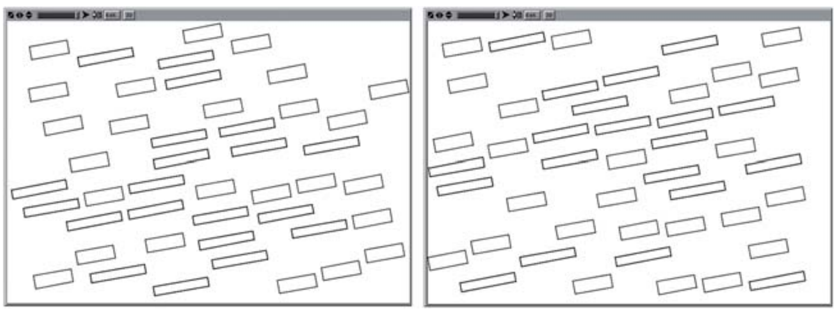

Figure 6. Two results of the same parameters shown in figure 5.

4.1. ACCUMULATIVE SHADOW DURATION

The sunshine duration of a building is actually the sunshine duration of the ground where the building is located. The sunshine duration of a piece of ground of a certain day is the sunshine duration of the day minus the duration that building(s) cast shadow on it. To calculate the shadow duration of the ground, we divide it into grid and divide the valid sunshine time into short periods, then calculate the accumulative shadow duration of each grid unit.

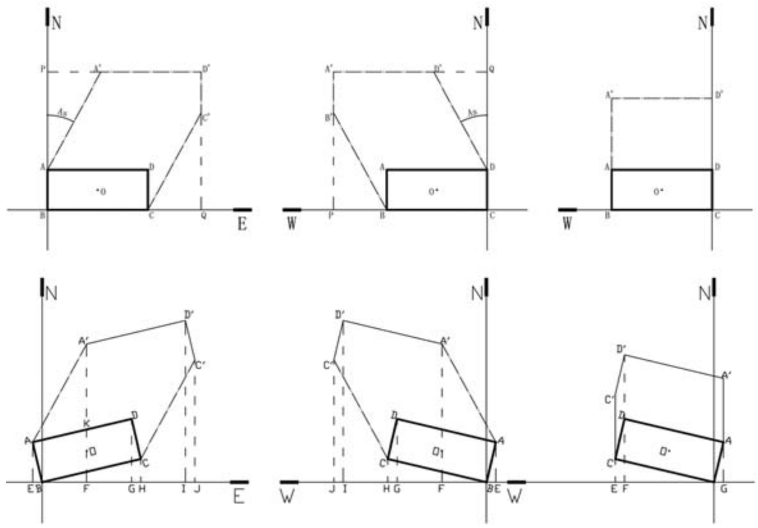

Figure 7. The shadow area casted by a building

The shadow area cast by a building at a certain time can be calculated through sun's azimuth and altitude, as well as the dimensions of the building. The question then becomes one of judging whether a point (patch) is inside a polygon (shadow) at a certain time.

As figure 7 shows, the shadow area is the concave Polygon AA′D′C′CD or ABB′A′D′D. In order to reduce the difficulty, we subdivided the concave polygon into two convex polygons, for example, dividing AA′D′C′CD into AA′D′D and DD′C′C. To determine a point inside or outside a convex polygon, first calculate the vector distances of the point to each edge, then compare the signs of the vector distances: only if all are positive or all are negative, the point is inside the polygon.

We use the patches' attribute Color to record the calculation result of each time period. Firstly we initialise the Color with a certain value. After each calculation, the value of Color of every patch that is inside shadow will be added 0.1, if the time period division is 10 minutes. At last the Color of the patches represents the accumulative shadow duration of the ground.

4.2. ACCUMULATIVE SHADOW DURATION CAUSED BY MORE THAN ONE BUILDING

When the shadow areas caused by more than one building overlap, the shadow duration of a patch cannot be simply accumulated. If the two buildings cast shadows on the same patches at the same time, the shadow duration of the patches should be added once rather than twice, otherwise it will cause error.

To solve the problem, we add the variant Shadow-Time for each patch to record the time list of shadow. When calculating the shadow caused by a building of a certain time, if the time list of a patch already contains the time, whether the patch is in shadow will not be calculated so that the total duration of the patch will not be added incorrectly.

4.3. POSITION-SEARCHING PROCESS

After the overall calculation, the housing buildings will check the total sunshine duration of the patches underneath to decide whether to enter the Position-Searching process, which is similar to that introduced in Section 2.

Because the calculation of the accumulative shadow duration of each patch is relatively slow, position-searching process will take a long time if it starts from a randomised distribution of housing buildings, which is the situation when the system starts. Hence after the initialisation, we start a procedure that the buildings move to make the distance between them bigger than the minimum fireproofing space, followed by the Position-Searching process for satisfying the constraint of sunshine duration.

4.4. PARAMETERS, INTERFACE AND RESULTS

The system of automatic planning under the constraint of sunshine duration has similar parameters to those under the constraint of sunshine spacing. The constraint of sunshine duration in current version was set as constants in the programming according to the condition of Nanjing, China.

Figure 8 shows the interface, and the usage is similar to the previous system. At anytime the user can click the “SUN-Calculate” button to calculate the sunshine situation of the time. Clicking the “MARK-SUM_TIME” button will shade the patches in color according to their sunshine duration. Figure 9 shows two results of the same parameters shown in figure 8.

5. Further work

The numerous and complicated results presented by the model based on the idea of multi-agent simulation are the outcomes of actions and interactions of agents. Such system reflects the complex phenomena of the real objects, whereas the model does not need to be complex. This idea related to complex systems is able to jump out of the increasingly complex analytical framework, indicating a new way of thinking. Similarly, the automatic problem-solving for residential planning is difficult to achieve using the conventional idea of analytical calculation, while in multi-agent simulation framework; the model can be established with relatively simple mathematical approaches.

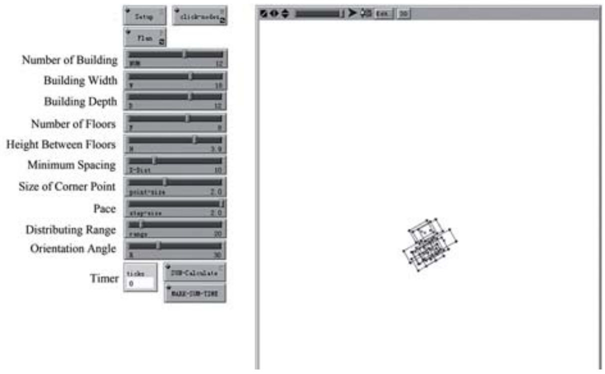

Figure 8. The interface of the system of automatic planning under the constraint of sunshine duration.

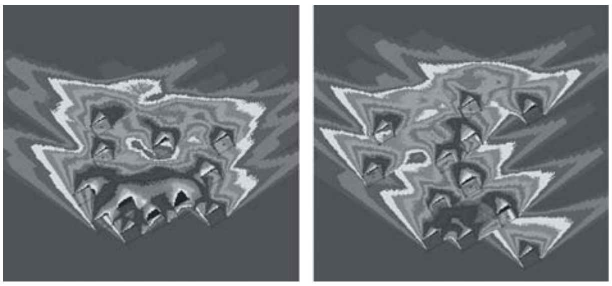

Figure 9. Two results of the same parameters shown in Figure 8, with colour shading indicating the time duration.

What we have shown is merely a test of computer-generated design, which applied the multi-agent simulation platform NetLogo to make automatic planning of residential quarter under insolation conditions. The establishment of data structure, the optimisation of algorithm, and the realisation of program in transforming architectural problems to computational models has been our key concern. It represents just a basic framework, in which a number of problems, such as the non-parallel layout of buildings, the shape of site, the interplay of the constraints of sunshine spacing and duration, the mixture of various planning objects, and so forth, have yet to be considered. Additionally, the algorithm to calculate shadows, which is very crucial for the system speed, will be our next focus in the research. Moreover, the buildings as agents in our system now only have the behavior of position-searching. Their orientations, height and other attributes are still parameters to be set by the user. Our further work will add behaviors of the agents so that they can change those attributes in the problem-solving process. Finally we hope the system will be developed as application software that can be used in the practice of residential planning.

References

Batty, M.: 2003, Agent-based pedestrian modelling, available from: Open Source Repository < http://www.casa.ucl.ac.uk/publications/workingpapers.asp> (accessed April 2007).

Caneparo L., Masala E. and Robiglio M.: 2005, Dynamic generative modelling system for urban and regional design, *The 11th International Conference on Computer Aided Architectural Design Futures*, Vienna, 259–268.

Castle,C. J. E. and Crooks, A. T.: 2006, Principles and concepts of agent-based modelling for developing geospatial simulations, available from: Open Source Repository <http://www.casa.ucl.ac.uk/publications/workingpapers.asp> (accessed April 2007).

Krause, J.: 1997, Agent generated architecture: design and representation, *ACADIA '97*, Cincinnati, 63–70.

Lechner, T., Watson, B., Wilensky, U., and Felsen, M.: 2004, Procedural city modelling, available from: Open Source Repository <http://ccl.northwestern.edu/papers/> (accessed April 2007).

Lechner, T., Watson, B., Ren, P., Wilensky, U., Tisue, S. and Felsen, M.: 2006, Procedural modeling of urban land use, available from: Open Source Repository <http://ccl.northwestern.edu/papers/> (accessed November 2009).

Li, B.: 2007, A generic house design system: expertise of architectural plan generating, *CAADRIA 2007,* Nanjing, 191–198.

Li, B., Li, R., Xue, K., Liu, C., and Gao, Q.: 2008, A generative tool base on multi-agent system: algorithm of "HighFAR" and its computer programming, *CAADRIA 2008,* Chiang Mai, 335–342.

Testa, P., O'Reilly, U.-M. and Greenwold, S.: 2000, AGENCY GP: genetic programming for architectural design, eternity, infinity and virtuality in architecture, *The 22nd Annual Conference of the Association for Computer-Aided Design in Architecture*, Washington, D. C., 227–231.

Wilensky, U.: 1998, NetLogo scattering model, available from: <http://ccl.northwestern.edu/netlogo/models/Scattering> (accessed April 2007).