



# Demand-Driven Distributed Adaptive Space Planning Based on Reinforcement Learning

Jiaqi Wang<sup>(✉)</sup> and Wanzhu Jiang<sup>(✉)</sup>

School of Architecture, South China University of Technology, Guangzhou, China  
{ucbqwj0,ucbq121}@uc1.ac.uk

**Abstract.** In the second digital turn, the architecture driven by big data logic is gradually shifting from a traditional static entity to an intellectual living organism. This paper explores a space planning algorithm that applies reinforcement learning to the multi-agent system to achieve condition adaptability. This algorithm contains an inclusive environment and programmable agents that represent independent spaces. Through reinforcement learning, personalized space needs are quantified as the agent's Space Schema, which can provide adaptive behavior strategies to adjust volumetric room boundaries. The spatial organization emerges in multi-agent competition, guided by the Negotiation Schema, realizing the dynamic equilibrium of spatial relations and the stable maximization of collective interests. Through real-time interaction and distributed decision-making, this bottom-up method defines a new architectural paradigm that continuously changes based on demands with its high degree of variability, adaptability and evolvability.

**Keywords:** Demand adaptability · Space planning · Multi-agent system · Distributed decision-making · Deep reinforcement learning

## 1 Introduction

We live in an era with excess data, characterized by dynamism and complexity. Our environment is no longer seen as fixed, or shaped by forces beyond our control, but as in constant and noticeable change [2]. On the one hand, the changes in natural and social environments lead to unstable site conditions. On the other hand, the trends of diversification and customization in spatial needs result in various design requirements. However, architects always regard the environment and users as constant factors and acquiesce to the static nature of spaces. To cope with this stereotype, we need to consider the real-time adaptability of architectural design. While realizing automatic generation, architecture also needs to interact with designers, users, and environments to achieve a state of autonomy.

It is possible to realize the above goals in the second digital turn, especially by applying generative design methods and artificial intelligence in architecture. Under this influence, the architecture system can form intelligence to cope with changes in external

---

These authors contributed equally to this work and should be considered co-first authors.

© The Author(s) 2023

P. F. Yuan et al. (eds.), *Hybrid Intelligence*, Computational Design and Robotic Fabrication,  
[https://doi.org/10.1007/978-981-19-8637-6\\_23](https://doi.org/10.1007/978-981-19-8637-6_23)

conditions, realizing the transition from small data logic based on deterministic formulas to big data logic according to mass computation.

This research is part of the frontier study on autonomous architecture systems to cope with the ever-changing environment and complex design requirements. It aims to combine multi-agent system generation methods and machine learning technology to endow the digital architecture system with autonomy, establishing the Distributed Adaptive Space Planning Algorithm (DASP) that can respond to changes in diverse design problems in real-time and adaptively reconfigure space conditions and organizations. As an intelligent architectural framework, this bottom-up responsive design generation method includes the needs and wishes of users, the coordination and integration of AI, and the thoughts and notions of designers, forming a hybrid Intelligence (Fig. 1). Through distributed decision-making, it can maximize group goals while meeting individual needs, defining a new paradigm of architecture that adapts to human willingness continuously.

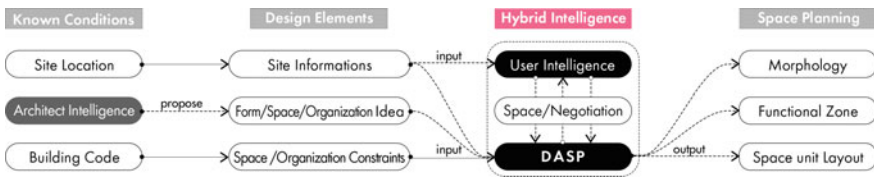


Fig. 1. The space planning process based on the DASP method

## 2 Literature Review

Space planning is a main task of architectural design: organizing space elements appropriately to meet a set of standards or achieve certain purposes, whose automation is what computational design hopes to achieve initially. However, the non-determinism of space planning is one of its main contradictions with computational methods, mainly manifested in (1) Design elements are discrete, unstable, and unstructured; (2) Incomplete design constraints prevent a single solution; (3) Conflicting design goals indicate there is no perfect solution; (4) Design evaluations are influenced by subjective styles and preferences.

The multi-agent system is an appropriate computational model to deal with indeterministic space planning problems. Its application can be divided into two types: the physical system in which agents represent space units, and the living system in which agents occupy space voxels. Guo [1] adopted three shapes of agents to represent spaces with different functions, whose positions are organized by attraction and repulsion forces, guiding the multi-story building generation. Meyboom [4] controlled the agents to occupy or release voxels through the principle of stigmergy. The agents interact indirectly with pheromone to arrange the position and shape of each space and form the building’s spatial layout. For these two algorithm models, the former has a simple framework and better controllability, but its insufficient organizational accuracy

leads to the low information capacity and distributed decision-making ability. In contrast, the advantage of the latter model lies in the interactivity and extensibility of its framework, which can add diverse spatial and environmental requirements. But due to the weak controllability of its agent behaviors, the planning results are difficult to echo with traditional spatial forms. Thus, this research tends to explore more operating rules and organizational forms of agents based on the latter model, focusing on agents' control strategies for diverse spatial requirements.

Faced with the intractable problems of indeterministic algorithms, the implantation of reinforcement learning is effective. As one of the three basic machine learning paradigms, reinforcement learning is a process of training the agent to continuously adjust its own strategy to take the optimal action to maximize the accumulated reward through interactions with the environment. The current application of reinforcement learning in CAAD focuses on topics of intelligent control [3, 5] and machine feedback [8], which also shows great potential in space planning research. NoMAS [6], a project in UCL, trained a library of space modules as agents which can be reconfigured according to user needs, developing an algorithm model to generate the residential complex. Pedro Veloso [7] trained a set of spatial agents to autonomously complete custom spatial configurations and interact with designers in real-time to generate floor plans. In the above work, reinforcement learning endows the agent with specific behavior patterns when dealing with the design target. This study also implants it into the multi-agent system to enhance their control, training agents to coordinate different spatial demands and respond in real-time.

### 3 Methodology

Distributed Adaptive Space Planning (DASP) takes the multi-agent system as a framework, adopting various methods including reinforcement learning and stigmergy principle, comprehensively combining bottom-up local agents with top-down global constraints. It connects users and the environment to accommodate dynamic activity needs, diverse design specifications, and critical site conditions, becoming an interactive, adaptable, and expandable space planning method (Fig. 2). The process can be divided into four steps: (1) the construction of the planning environment; (2) the representation of the space agent; (3) the implantation of the Space Schema; (4) the implementation of the Negotiation Schema.

#### 3.1 Planning Environment

DASP uses a 3D grid formed by discrete spatial voxels as an inclusive planning environment. Each node's size is determined by the spatial scale of the agent and the resolution required for planning. In this experiment, 0.3 m is selected as the side length of the basic unit to respond to the ergonomic scale and acquire the refined planning results. Each vectorized point in this matrix has two basic information of position and color, where the initial color is ( $r = 0, g = 0, b = 0$ ), which can be dyed by the agent and its pheromone. In addition, each node can also be assigned multi-level environmental information, imported in the form of formulas or bitmaps, and superimposed on each voxel.

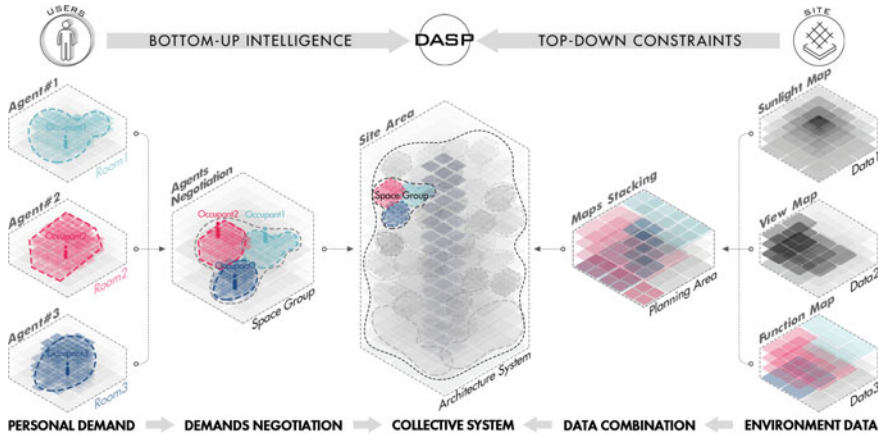


Fig. 2. The basic framework of DASP: integration of local agents and global constraints

This preliminary pheromone that guides the activities of agents can be transformed into a data set of [0, 1] in proportion as the top-down global constraints.

Different global constraints are divided into two categories in DASP, including masking-stack and mapping-stack (Fig. 3). The data in the masking-stack demarcates the areas that do not participate in the planning, including terrains and obstacles, etc. The mapping-stack contains physical environment maps (sunlight maps, wind maps, noise maps) and social environment maps. The conversion ratio ( $T_E$ ) of different types of data ( $E$ ) can be set according to empirical research or local requirements and unified into the demand degree of the same measurement ( $E_D = E \times T_E$ ). These global constraints can be used as a manifestation of group consciousness, setting the collective goals of the multi-agent system and preventing its autonomous behavior from falling into chaos.

<b>MASKING-STACK</b>	<b>0/1</b>
<b>Environmental Masking</b>	[X,Y,Z]
[Special Terrain] Slope / Stage / River	
[Reservation Area] Trees / Heritage Buildings	
[Obstacle Area] Rocks / Other Buildings	
<b>Architectural Masking</b>	[X,Y,Z]
[Structure Area] Frame / Core-Tube	
[Circulation Area] Stairs / Elevators / Corridors	
[Green Area] Atrium / Garden	
<b>MAPPING-STACK</b>	<b>0-1</b>
<b>Physical Environment Map</b>	[X,Y,Z]
[Natural Data] Sunlight / Wind / Temperature	
[Living Data] View / Noise / Traffic	
[Special Data] Entrance / POI	
<b>Social Environment Map</b>	[X,Y,Z]
[National Data] Pandemic / Policy	
[Local Data] Festival / Traditional Custom	
[Special Data] POI	
<b>Architectural Map</b>	[X,Y,Z]
Privacy - Public / Main - Auxiliary / Special Zone	

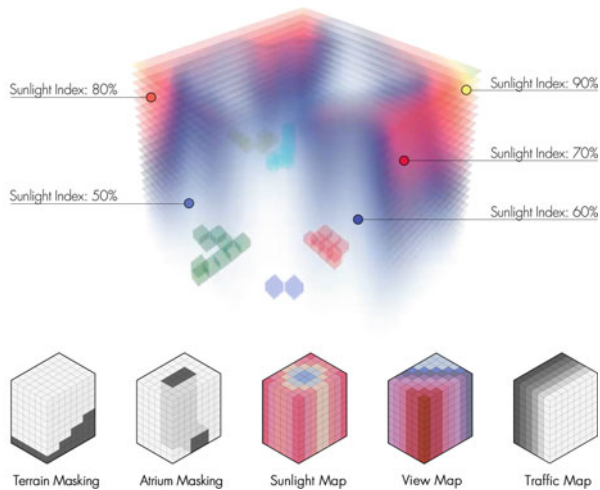
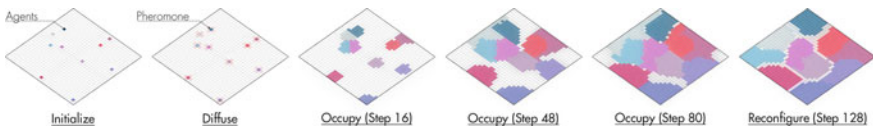


Fig. 3. Global constraints, left: detail parameters of two stacks, right: some examples

### 3.2 Space Agents

An agent representing a user space is embodied as a point cloud collection of nodes with a particular shape in the environment. In the beginning, the agent will be initialized as a cell at a specified position, following preset rules to occupy or release voxels in the environment and gradually forming a closed geometry, that is, the expected space. When the algorithm runs, the agent will emit pheromone and receive pheromone from adjacent agents and the environment to perceive the surrounding state and determine the expansion behaviors. In addition, two agents cannot occupy the same node simultaneously but can negotiate and reconfigure the occupied area. In this framework, if the design requirements and specifications are properly translated into the operating rules of the agent, it is possible to simulate any spatial state and its combination in real-time (Fig. 4).



**Fig. 4.** Planning process of the agents

Each agent in DASP has the following basic parameters.

1. The color (R, G, B) is defined as a user hue, which can link the user characteristics. These parameters are obtained from questionnaires, which can provide personalized information and help organize the agent relationships.
2. The pheromone carries color information, starting from the center of the occupied shape, picking up the three-dimensional Von Neumann Neighborhood iteratively, and diffusing outwards to realize the gradual contagion of the nodes.
3. The capacity represents the upper limit of the node number that the agent can accommodate, corresponding to the space volume. For example, a standard room of  $6\text{ m} \times 3\text{ m} \times 3\text{ m}$  contains 2000 voxels with a side length of 0.3 m. When the occupied nodes reach capacity, the calculation will continue, and the agent can update the occupied area by releasing internal voxels and occupying new ones.
4. The expansion rules guide the agent to select suitable occupation voxels, controlling the growth mode and the generation result. Each adjacent node ( $P_N$ ) of the occupied area  $P$  is substituted into the customizable function  $F_t(x, y, z)$ , and the node ( $P_{N_i}$ ) with the smallest or largest result ( $F_t(P_{N_i}) = F_t(P_N)\text{min/max}$ ) is chosen as the next occupied one. Based on the different design objectives, various calculation methods can be set to promote diverse growth behaviors.

### 3.3 Space Schema

Each agent is assigned local identities to develop a real-time adaptive connection between abstract agent behaviors and specific user needs. The concept of Schema is introduced into DASP as an agent attribute, which reflects its instinctive behaviors in response to condition changes or external disturbances.

The significance of the Space Schema (Fig. 5) is to enable the agent to generate a specified space according to personalized requirements. It employs three control elements of volume, proportion, and form. The volume determines the spatial scale, the proportion indicates the aspect ratio, and the form describes the particular shape. The setting of the Space Schema realizes a comprehensive description of a space and ensures that the user’s demand can be transformed into a combination of several parameters to direct the agents’ planning behaviors.

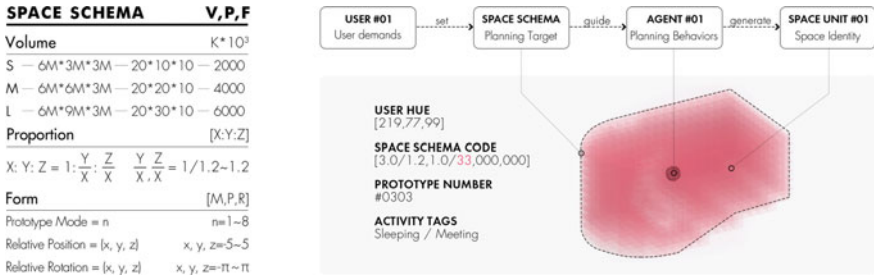


Fig. 5. Space schema, left: detail parameters, right: principle and example

This study innovatively introduces reinforcement learning as a decision-making mechanism to embed the Space Schema into the agent (Fig. 6). In the experiment, the randomness of V, P, and F is limited to preliminarily verify the method’s feasibility. The space volume is set to 2–4 ( $\times 10^3$ ) voxels, the rate(y/x) and rate(z/x) choose 5 values from  $1.2^{-1}$  to 1.2, and the form takes 8 geometries of different themes, generating a total of 600 parameter combinations. Before each episode starts, the system will initialize a target schema in the planning environment according to a random parameter set.

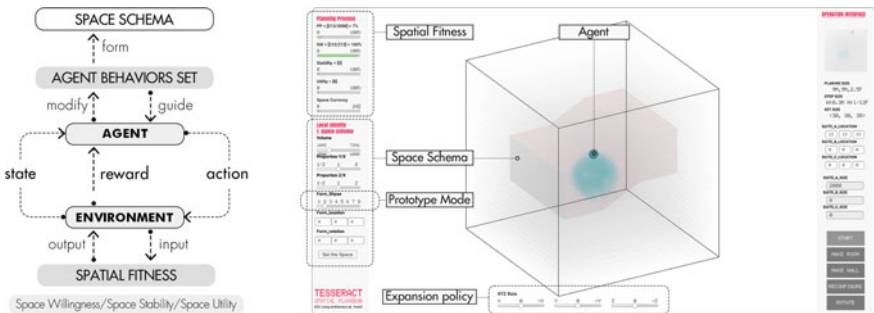


Fig. 6. Left: the principle of reinforcement learning, right: the computational model

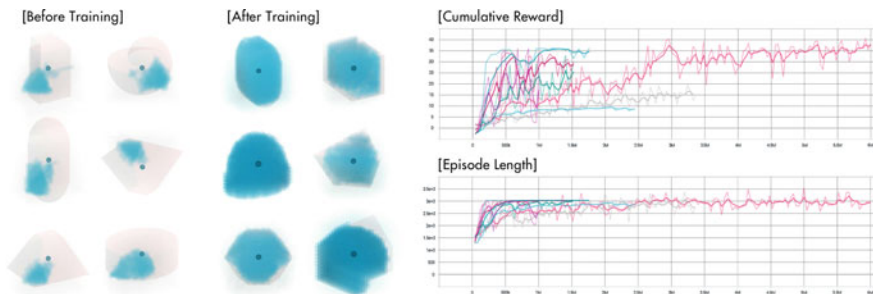
In the training framework, the discrete actions were set to 7 to separately increase or decrease the values in the policy (x, y, z) to decide the agent’s expansion function and adjust its growth direction and intensity in steps.

The observation contains the information for planning goals and current status. In this experiment, it consists of these parameters: (1) the index of Space Schema as the

target status. (2) the agent current state, including its centroid position, the last occupied point position, and the expansion rules (x, y, z). (3) the current planning state, including all the points in the occupied area.

There are three types of rewards in the experiment: (1) real-time rewards, each time the expansion points are inside the target schema, a reward will be added. (2) Periodic rewards, when the number of points that meet the requirements in the agent-occupied area accounts for a certain value (30%, 60%) of the agent capacity (this value is called space willingness), real-time rewards will increase. (3) Target rewards, when the agent reaches capacity and the space willingness exceeds 90%, the planning is completed, and the maximum reward is obtained. We also set up rewards based on structural stability and spatial practicality analyses.

After  $6 \times 10^6$  times of training, the training curve is stable at the maximum value of 36. In this case, the agent can substantially adapt to the changes in the Space Schema in real-time and complete the corresponding space planning tasks (Fig. 7).



**Fig. 7.** Left: The behaviors of the agents after training, right: graphs (pink one success) with the cumulative reward and the episode length of the agents (y-axis) over the episode (x-axis)

### 3.4 Negotiation Schema

Negotiation Schema is another local identity of agents, representing the tendencies in the space organization. It is mainly responsible for regulating the agent's relationships with its neighbors and environment to endow the whole system with adaptability. The setting of the Negotiation Schema (Fig. 8) aims to realize the agent's response to the user's organizational willingness, completing the collective space arrangements through deformation or displacement.

In DASP, the Negotiation Schema consists of three sets of parameters, which describe the organizational relationship at different scales. Among them, the interaction tendency represents the adjacency willingness of the agent with surrounding agents. According to the propensity ranking of “away–reject–none–accept–adjacency”, the set of weights  $K_I$  contains a set of numbers  $\{k_{i1}, k_{i2}, k_{i3} \dots\}$  in the range  $[-2, 2]$  for each surrounding agent. In the algorithm, this relationship is encoded in a group of attractors, which generate attractive or repulsive forces according to the positive or negative values of  $K_I$ . These forces make agents away from or close to each other and weaken as the distance





Fig. 8. Negotiation Schema, left: detail parameters, right: principle and example

increases. Each agent will experience a combined force from neighbors to adjust its morphology and steer it toward a suitable social position.

Cluster tendency is more inclined to describe the grouping willingness based on the agent’s color. The user can set the tendency weight  $K_C$  to similar color groups in the range of  $[-1, 1]$ , which is multiplied by the pheromone value in the planning area and directly acts on the expansion rule of the agent, to lead the agent to gradually move toward or away from the group in a specified color.

Environment tendency quantifies the agent’s willingness for environmental resources. The set  $K_E$ , which is influenced by space function and user preferences, includes multiple weights between  $[-1, 1]$  corresponding to indexes in global constraints. In this way, the agent’s comprehensive environmental satisfaction degree for a certain node can be obtained quantitatively through a weighted average method. Therefore, the environmental resources of the whole system can be reasonably allocated to maximize the utilization rate.

The Negotiation Schema coordinates the agents’ relational network and further optimizes the spatial form generated by agents, which greatly expands the adaptability of DASP when dealing with complex demands.

### 4 Experiment and Application

We validated the space planning potential of DASP when dealing with complex real-time demands in practice through several sets of experiments. First, the simulated daily schedule of an occupant guides the Space Schema for parameter modification. Figure 9 shows the different spatial forms generated by the agent in this continuous planning. When the Space Schema is changed in this experiment, the DASP algorithm can autonomously remove the unnecessary nodes and perform the space planning according to the adjusted parameters. However, for some shapes with acute angles, such as the space at 19:30, the error is still relatively large, related to the precision setting of the agent’s expansion rule.

The second experiment was carried out in a range of  $12\text{ m} \times 12\text{ m} \times 6\text{ m}$ . By placing five agents in this area, the possible negotiation activities of two users in this system were simulated to generate a space group. By adjusting the interaction tendency, we simulated the gradual process of the organizational relationship between two users from no tendency to mutual resistance and mutual acceptance. The parameter changes of the Negotiation Schema trigger agents’ various behaviors such as deformation, attraction,



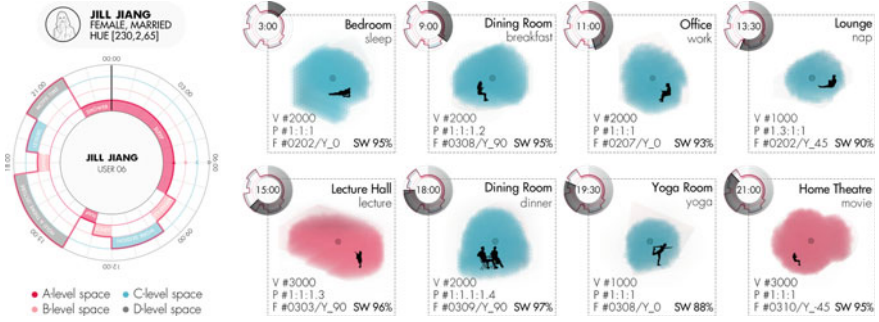


Fig. 9. Experiment 1, left: simulated timetable, right: the space sequence and its parameters

and repulsion. Thus, the space occupancy is negotiated, and different space forms emerge (Fig. 10). Obviously, the growth of the five agents in the first stage does not affect each other, tends to be far away in the second stage, and begins to approach in the third stage.

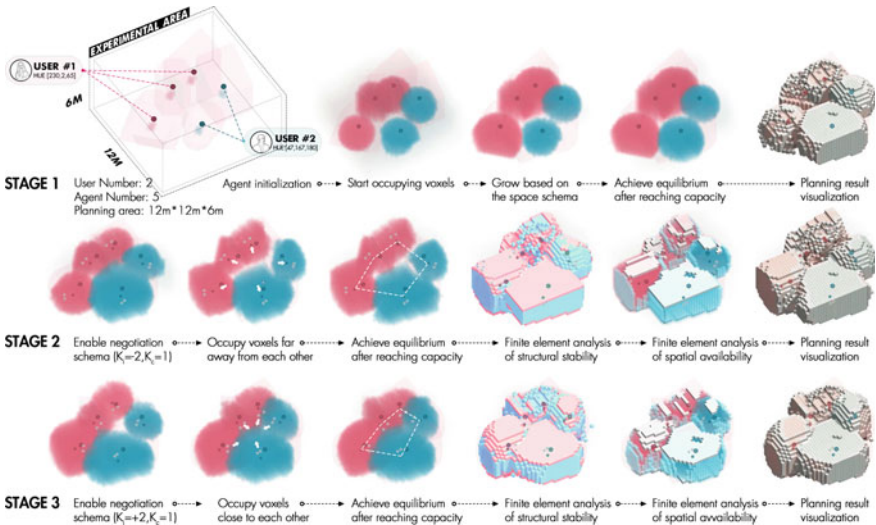
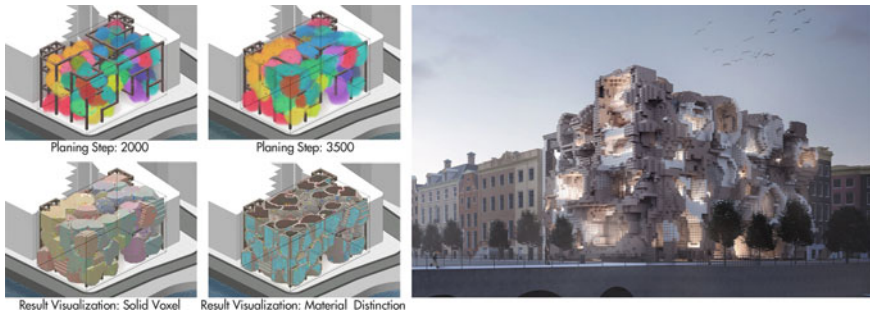


Fig. 10. Experiment 2, the planning process of three stages and their planning results

Finally, this algorithm is applied to developing the adaptive living complex, TESSERACT (Fig. 11). This project uses DASP as the core space design and generation tool, cooperating with several well-designed subsystems like the interactive platform and robotic material. Relying on the DASP’s adaptability, interactivity, and expandability, TESSERACT defines a new paradigm of living architecture that adapts to the human will and changes continuously.



**Fig. 11.** Project TESSERACT generation, visualization and rendering

## 5 Conclusion and Future Work

This research proposes an interactive algorithm DASP with demand adaptability, using reinforcement learning to establish a spatial solution that can respond to changes in user needs and environmental conditions in real-time. DASP implants Space Schema, Negotiation Schema, and global constraints to the multi-agent system as linkages to users, communities, and environments, forming a distributed decision-making method. The experiments show that trained agents can respond well to the parameter adjustment and perform adaptive behaviors, revealing the potential of DASP in autonomous space planning. The future work will focus on the limitations of DASP. First, the spatial organization purely based on social relations and user regulation may lead to space configuration problems. Other architectural constraints like topological relations should be considered. Second, the adaptability now totally depends on input prototypes. It is necessary to continuously modify preset prototypes based on user interaction, and guide the transfer of agent's behavioral patterns, realizing the evolution of the Space Schema. Also, using it as a design or analysis tool for architects is a new direction.

**Acknowledgements.** Project TESSERACT is conducted in UCL B-Pro AD RC3, supervised by Tyson Hosmer, Octavian Gheorghiu, Philipp Siedler, and Ziming He, and developed by Jiaqi Wang, Wanzhu Jiang, Ying Lin, and Zongliang Yu.

## References

1. Guo Z, Li B (2017) Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system. *Front Archit Res* 6:53–62
2. Hanna S (2020) Architecture as agent. Georgia Institute of Technology School of Architecture
3. Hosmer T, Tigas P (2019) Deep reinforcement learning for autonomous robotic tensegrity (ART). In: Proceedings of the 39th annual conference of the ACADIA, Austin, Texas
4. Meyboom A, Reeves D (2013) Stigmergic space. In Proceedings of the 33rd annual conference of the ACADIA, Cambridge
5. Smith S, Lasch C (2016) Machine learning integration for adaptive building envelopes. In: Proceedings of the 36th annual conference of the ACADIA, Ann Arbor

6. Tigas P, Hosmer T (2021) Spatial assembly: generative architecture with reinforcement learning, self play and tree search. [arXiv:2101.07579](https://arxiv.org/abs/2101.07579)
7. Veloso P, Krishnamurti R (2020) An academy of spatial agents: generating spatial configurations with deep reinforcement learning. In Proceedings of the 38th eCAADe conference, vol 2, Berlin, Germany
8. Xu T, Wang D, Yang M, et al (2018) An evolving built environment prototype. In: Proceedings of the 23rd CAADRIA conference learning, adapting and prototyping, vol 2, pp 207–215

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

