ON SLIME MOLDS AND CORRIDORS

The application of network design algorithms to connect architectural arrangements.

P. VELOSO, R. KRISHNAMURTI
Carnegie Mellon University, Pittsburgh, PA - USA
Email address: pveloso@andrew.cmu.edu
Email address: ramesh@andrew.cmu.edu

Abstract. The use of adjacency graphs to represent and generate architectural arrangements tends to favor direct connections between contiguous rooms. These disregard specialized circulatory systems (such as corridors), which consider connections between non-contiguous spatial units or accesses. This paper addresses two specific issues: (1) how to represent a circulation network for a specific access/adjacency graph embedding; and (2) how to design good circulatory solutions for the arrangement that optimizes this network. To represent a complete circulation network, we propose a scheme, an adapted straight skeleton, based on the boundaries of the spatial units. To design possible circulation alternatives, we adopt the Slime Mold model (Tero, Kobayash and Nakagaki, 2006, 2007). Using this model, we develop an original method, termed Adjacency Graph Selection (AGS), to generate circulation solutions for arrangements. As an initial test case for AGS, we use floor plan of the Louvre Abu Dhabi, designed by the French architect Jean Nouvel.

1. Introduction

In the field of computational design, one of the most prominent structures to analyze and generate spatial arrangements for buildings are networks. Graph theory specifies vertices (or nodes) and edges (or connections) as the basic elements to represent a network. Different approaches to graphs have been established to deal with different aspects of architectural arrangements. From the pioneering work of Lionel March and Philip Steadman (Steadman and March, 1974, Steadman in March, 1976, Steadman, 1983), we can cite three types of graphs for spatial representation. Plan graphs encode the physical compartment of the building such that edges are walls, and nodes are corners. Adjacency graphs represent the proximity of spatial units in plan – nodes are the spatial units while edges connect spatial units that should be close or even share a wall. Lastly, access graphs represent the real connection between spaces. It is a subgraph of the adjacency graph as the nodes represent spaces and edges represent only those adjacent spaces connected by doors or openings.

In particular, these connective representations of spatial units (adjacency and access graphs) have been widely influential in space planning research. The early pioneers considered the data structure of these graphs as an architectural representation through which to explore possible graph embedding – that is, a particular drawing of a graph – and its translation to arrangements. Current research examines the adjacency/access graph through different computational techniques such as physical simulation or agent-based modeling, in order to explore solutions dynamically.

2. The problem

Despite the potential of the adjacency graph as a representation for space planning, it is still too abstract to represent the geometry of an architectural circulation system. There is one important difference between configurations of corridor systems in architectural arrangements when compared to other systems, such as transportation or biological networks. Corridor systems establish indirect connection between individual spatial units. One historical explanation is that over the centuries, in complex buildings, the *enfilade* system (sequences of interconnected rooms) has given way to a network of corridors, enabling each room to exert a specific function and to keep a sense of privacy (Evans, 1997). Therefore, the distinction of specialized rooms and specialized circulatory systems is problematic for the generation of spatial arrangements based on an adjacency/access graph representation. Different computational techniques can result in satisfactory embedding for the graph. However, the spatial configuration of the embedding can indicate only direct access between contiguous rooms, not the possible corridors.

To address this problem, we assume the existence of a set of well-defined spatial units connected by an adjacency graph embedding, which is neither necessarily planar nor devoid of crossings between its edges. Spatial units that are supposed to be connected are not always adjacent, due to a dispute with other spatial units or even due to external constraints such as visibility, orientation, lighting, etc. In this case, the circulation network should ensure that the pair of spatial units connected by the adjacency graph has a good and efficient corridor system among them.

Our method provides a solution to two questions: (1) how to represent all possible and feasible circulation elements of these spatial units; and (2) how to generate good circulatory solutions from subsets of this complete network, based on the adjacency graph.

3. The method

In this paper, we adopt a constructive research method by developing a computational design approach that addresses the limits of the adjacent graph representation, which we then apply to a test case in order to evaluate its initial feasibility. The method addresses early iterations in the design process. It assumes that there might not be sufficient information for complex simulations and that the exploration of alternatives and opportunistic structural changes be pervasive. The method, therefore, should allow for on-the-fly changes to the initial parameters, and should generate a diagram of the corridors properly adapted to different initial parameters without requiring additional steps. We now consider the two questions.

(1) To generate possible feasible circulation elements of the plan, we investigated techniques for developing from the positions of the spatial units a complementary network that considers the geometrical qualities of a circulation system. This network may not only contain *enfilade* connections, but also corridors between the boundaries of the spatial units.

(2) We investigated an algorithm that selects and dimensions a subset of all different possible connections in the complete network, in order to represent a circulation system. In essence, this is a form of combinatorial optimization guided by the adjacency graph and the parameters of the algorithm.

We looked at a specific project to test both steps, namely, the Louvre Abu Dhabi, designed by the French architect Jean Nouvel. It comprises an arrangement of galleries and services in which circulation operates with both *enfilade* rooms and a pier between all spaces, providing the opportunity to explore computationally different design alternatives. In the future, we will employ the method in the generation of new spatial arrangements.

4. Generating a complete network (CN)

As any potential corridor borders a spatial unit, the generation of a complete circulation network does not depend on the configuration of the adjacency graph, but on the actual positions of the spatial units and the shape of their boundaries.

In Rhino, Grasshopper and GH Python, we can extract a vector representation of the rooms of the Louvre Abu Dhabi. We cluster close and small rooms and apply a convex hull algorithm to ensure that all spatial units are convex (Fig. 1a-c). Then, we apply a Delaunay triangulation with a filter to generate a graph with the actual adjacency of the units (Fig. 1d). The Delaunay represents all direct connections between the units or, in other words, the complete *enfilade* network.

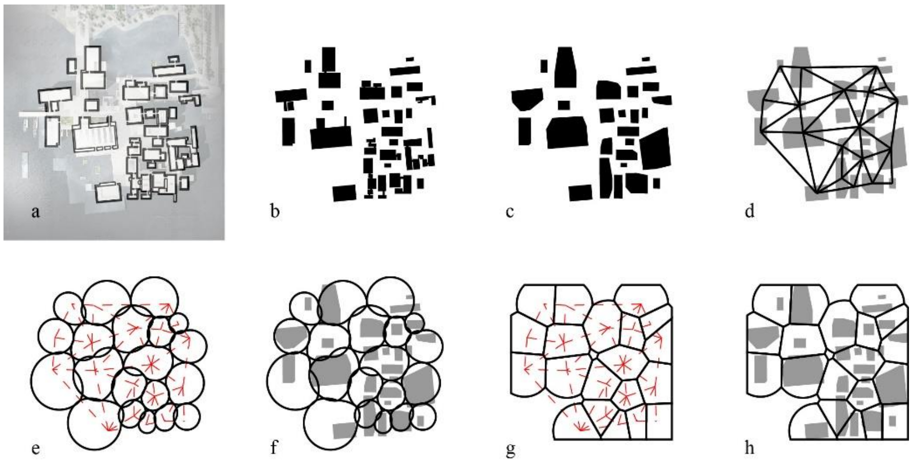

Figure 1. (a-c) Development of the spatial units; (d) Delaunay triangulation as *enfilade* system; (e-f) Pseudo-graph as corridor system; (g-h) Voronoi as corridor system

The next step is to produce the complete network of corridors between the existing spatial units. We tested two initial alternatives. The first is a pseudo-graph that connects the midpoints of the edges of the Delaunay graph with curves, generating a bubble diagram enveloping the original nodes (Fig. 1e-f). The second alternative is the dual graph of the Delaunay triangulation: the Voronoi diagram. It generates divisional lines in the midpoints of the Delaunay triangulation, forming polygonal cells around the original nodes with an optional control of maximum radius (Fig. 1g-h).

Both alternatives had the advantage of connecting the original triangulation in the midpoints (not all midpoints in the case of the Voronoi), forming a coherent combination of graphs. However, they are not sensitive to the geometric boundaries of the spatial units, generating a network of edges that would require extra steps to be translated to diagrammatic representation of corridors.

4.1. STRAIGHT SKELETON

For the final complete network, we adopted the straight skeleton. It consists of offset line segments between the pre-existing geometry, forming an internal or external skeleton connected to the original nodes by diagonals. We tested the implementations using kinetic triangulation, proposed by Palfrader, Held and Huber (2012) and the consolidated CGAL library.

In order to set the space between the spatial units and around the perimeter as the site for the skeleton, we defined an adaptive concave hull with all the units and then we offset it (Fig. 2a-b). After defining the external boundary (offset concave hull) and the internal boundaries (spatial units), the skeleton occupies all interstitial spaces for the corridors, forming cells (Fig. 2c). Then, the external boundary and diagonals of the skeleton are eliminated (Fig. 2d).

In order to define the accesses from the units to the corridor network, we defined two options. Multiple accesses use the intersection of the edges of the adjacency graph with the cells (Fig. 2e) to select the nodes representing accesses (Fig. 2f). This stimulates circulation through a room and even enfilade systems. Single access uses the same method to define the access point, however it selects only the edge of the adjacency graph connected to the largest neighbor unit (Fig. 2g).

The straight skeleton forms an adaptive network of polygonal cells adapted to the boundaries of the spatial units. It consists of centerlines that inherit the geometric complexity of the units. When there are more points and segments in the spatial units, the network will also have more segments. In our examples, the convex hulls of the spatial units are not completely orthogonal and the units have different sizes, orientations and alignments.

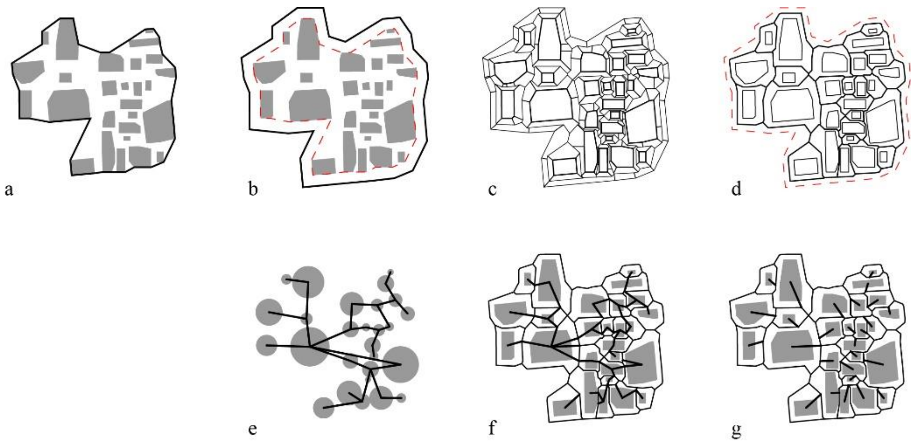

Figure 2. (a-d) Straight skeleton development; (e) adjacency graph; (f) multiple accesses per unit; (g) single access per unit

5. Let a slime mold design our corridors

With a developed complete network, the method next has to generate alternatives of the circulation systems. To address this issue, we looked at the Slime Mold (*Physarum polycephalum*), a unicellular organism that, in its vegetative phase, depends on a natural adaptive network of tubes (or edges) to find sources of food in the environment and then to distribute the nutrients and chemical signals around its body. The hydrostatic pressure due to rhythmic contractions of actomyosin fibers to transport nutrients increases the amount of flux that in return activates a shear effect on the tube, orienting the actomyosin fibers in the direction of the stretching force (widening the tube). As a result, dead end tubes and longer tubes tend to disappear while shortcuts between food sources are reinforced. In other words, it solves the combinatorial optimization problem using feedback between the flux of nutrient and thickness of the tubes.

5.1. THE SLIME MOLD MODEL

We adopted the model proposed by Tero, Kobayash and Nakagaki (2006). It interprets the behavior of the Slime Mold, using the Pouseuille flow principle associated with one food source and one sink, and avoiding pressure oscillation and rhythmic contractions. It comprises four steps:
(a) Assignment of an initial value for the conductivity (D) and length (L) for all tubes, and assignment of the constant of the flux from the source (I_0).

(b) Setting the sink pressure as zero and solving the system of Network Poisson equation to find the pressure of all other nodes.

$$\sum_i \frac{Dij}{Lij}(pi - pj) = -I_0 \; for \; j = 1; +I_0 \; for \; j = 2; \; 0 \; otherwise \quad (1)$$

(c) Using the Poiseuille flow formula to obtain the flow of the tubes.

$$Qij = \frac{Dij}{Lij}(pi - pj) \quad (2)$$

(d) Updating the conductivity of the tubes using the dimensionless form of adaptation equation associated with a monotonically increasing function.

$$\frac{d}{dt}Dij = f(|Qij|) - Dij \quad (3)$$

In order to optimize railroad networks, Watanabe, Tero, Takamatsua and Nakagaki (2011) modified the original model to operate with multiple food sources. They proposed three methods to select the points to be sources and sinks for each unit step. In Two Points Selection (TPS), the size of the food sources define the probability for the selection of two points for source and sink. In Multipoint Selection Method (MPS), the same food-based biased sampling defines *n* pairs of points. In Complete Multipoint Selection Method (CMPS), a combination of all the nodes define the pairs. These methods normalize the flow and the conductivity according the number of pairs selected at each step.

These methods can be adapted to design circulation networks, as parameters, type of function, and the food sources can control the output. However, there are still some limitation. TPS and MPS rely on probabilistic selection, which means that the output can vary (Fig. 3a and 3b) and converge to sub optimal results depending on the number of iterations and on the values of the food sources. CMPS combines all the elements of the network for each time step. It can be deterministic, but it is always computationally expensive.

5.1. ADJACENCY GRAPH SELECTION METHOD

The method proposed in this paper is Adjacency Graph Selection (AGS). We implemented it in Python and integrated it with the complete network developed in Rhino, Grasshopper and GH Python. We treat each spatial unit as a food source of the slime mold, dimensioned according its area. The complete network consists of tubes between nodes without food - potential candidates for corridors – connected to these spatial units (nodes with food). Instead of probabilistic selection (TPS or MPS) or complete combination (CMPS), AGS uses the adjacency graph to define the source-sink relationships. At each step, the selection consists of all paired combinations of one spatial unit with all its neighbors in the adjacency graph (as sinks). In total, it operates with a selection of two times the edges (2xE) of the adjacency graph for each time step.

This method avoids discovering a proper *n* values for probabilistic selection, avoids using repeated pairs and constrains the selection set only to desired connections. In combination with the settings of the original model, the adjacency graph is the device that allows the designer to control bias of the system towards specific optimum solution with an efficient selection of sources and sinks.

The initial results confirm that AGS can direct the development of the slime mold (compare Fig. 3c, 3d with 3e), but it does not follow the tendency of MPS (combined with our initial settings) to quickly define spanning trees. AGS accentuates the weight hierarchy of the corridors, suggesting even areas that look like halls or foyers. The filtering of the accesses also affects the type of solutions. Additionally, the results confirm that with multiple accesses, the slime mold tends to form *enfilade systems* (Fig.3c). With single accesses, the slime mold privileges a hierarchy of corridors (Fig.3d).

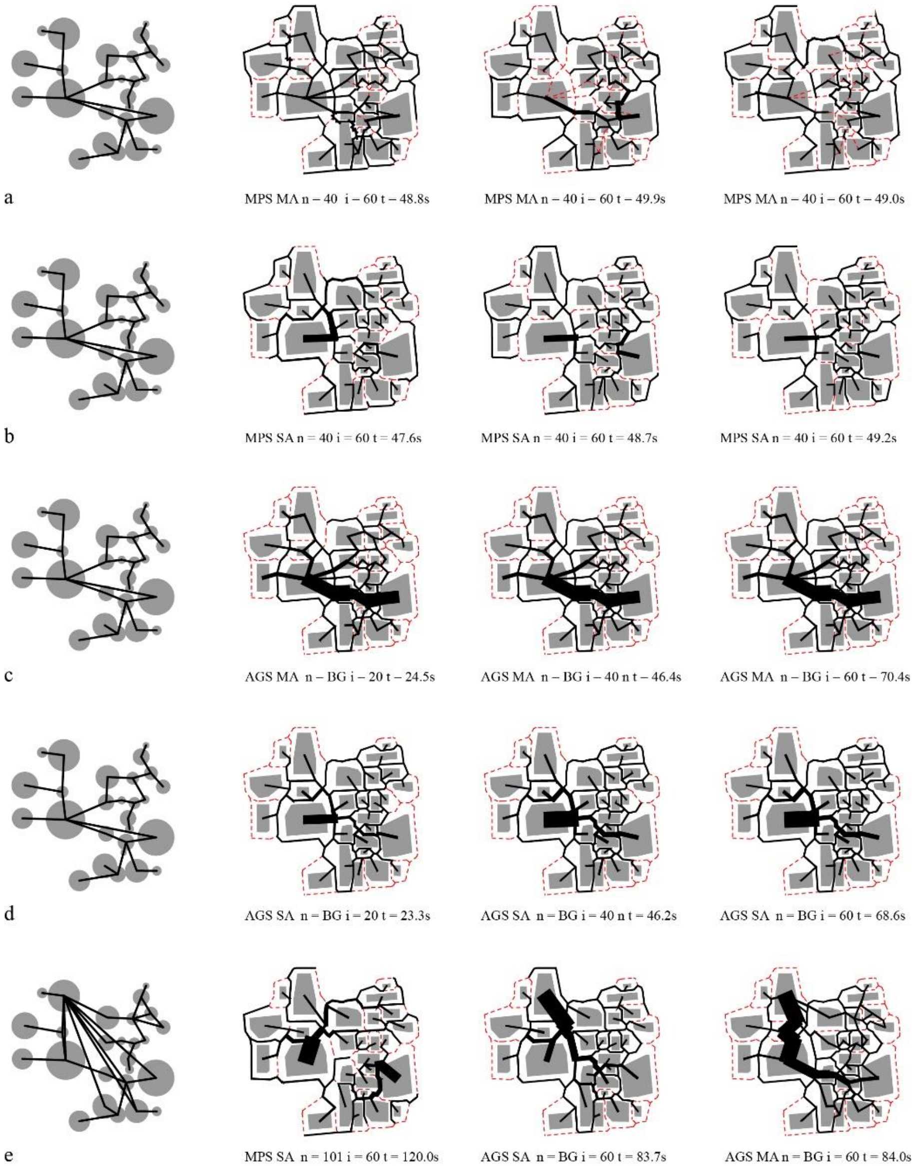

Figure 3. (a) Three variations of MPS and multiple access (MA), 40 samples per step and 60 steps; (b) Three variations of MPS and single access (SA), 40 samples per step and 60 steps; (c) A sequence of AGS and multiple access, samples defined by graph, 20, 40 and 60 steps; (d) A sequence of AGS and single access, samples defined by graph, 20, 40 and 60 steps; (e) A different graph with corresponding MPS + SA, AGS + SA, and AGS + MA

6. Conclusion and future steps

The initial results offer evidence that the proposed method is able to generate a complete network and to design new connections under the control of the adjacency graph. However, some limitations were identified. In future implementations, the filtering of edges for single access mode will occur during the optimization, in order to ensure that optimal corridors will concentrate the accesses. The method for multiple sinks and sources has to be improved in order to control the redundancy of the corridors and to converge to good alternatives. Another drawback is that current generation of the complete network and optimization are slow for a real time interaction. Future implementation will consider scientific computing methods to integrate it with a space planning interactive tool.

Acknowledgements

This research was supported by CNPq (National Council for Scientific and Technological Development). We also thank professors Ziv Bar-Joseph and Matt Ruffalo from the Computational Biology Department at CMU who provided insight and expertise with Bi-directional studies of natural networks.

References

EVANS, R., 1997. Figures, doors and passages. *In: Translations from Drawing to Building and Other Essays*. London: AA Publications, 55-91.

MARCH, L., AND STEADMAN, P., 1974. *The geometry of environment: an introduction to spatial organization in design*. 1st ed. Cambridge: The MIT Press.

PALFRADER, P., HELD, M. AND HUBER, S., 2012. On Computing Straight Skeletons by Means of Kinetic Triangulations. *In:* L. EPSTEIN, AND P. FERRAGINA, eds. *20th European Symposium on Algorithms*. Berlin: Springer, 766-777.

STEADMAN P., 1976. Graph-theoretic representation of architectural arrangement. *In:* L. MARCH, ed. *The Architecture of Form*. New York: Cambridge University Press, 94-115.

STEADMAN P., 1983. *Architectural Morphology: an introduction to the geometry of buildings plans*. London: Pion Limited.

TERO, A., KOBAYASHI, R., NAKAGAKI, T., 2006. Physarum solver: a biologically inspired method of road-network navigation. *Physica A* 363, 115-119.

TERO, A., KOBAYASHI, R., NAKAGAKI, T., 2007. A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology* 244, 553-564.

WATANABE S., TERO A., TAKAMATSUA A., NAKAGAKID T., 2011. Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism, Physarum plasmodium. *Byosystems* 105, 225-232.

TERO, A., TAKAGI, S., SAIGUSA, T., ITO, K., BEBBER, D. FRICKER, M., YUMIKI, K., KOBAYASHI, R. NAKAGAKI, T., 2010. Rules for Biologically Inspired Adaptive Network Design. *Science* 327, 439-442.