Multi-Stage Agent-Based Optimization of Building Layouts

David Rulff[1], Ralph Evins[1]
[1]Energy in Cities group, Department of Civil Engineering, University of Victoria, Canada

Abstract

Early decisions on building program and room topology influence built form and zoning, often imposing immutable constraints on system design while having a disproportionately large impact on performance. This paper explores analogous concepts and techniques from a variety of fields (including graph theory, architectural practice, machine learning and physics simulation) to develop a novel agent-based multi-objective method to rapidly generate feasible building layouts and converge on potentially optimal solutions. The problem is divided into three stages, each incorporating performance objectives and zoning constraints: room positioning (pseudo-physics-based dense packing of undirected, weighted graphs), layout generation (cellular growth), and performance refinement. An illustrative example for the first two stages is provided, with refinement options discussed at a qualitative level. Applying this method during the early conceptualization of buildings would support more integrated design processes targeting ambitious performance targets. The intention is to provide insight into the problem and directions for future work.

Introduction

During early concept design, architects commonly use bubble diagrams or adjacency matrices (prioritizing which rooms need to be near others) to represent the building program. These are typically manually generated, with the goal to visualize (not optimize) the information.

The body of architectural research includes a variety of methods for trying to generate and/or optimize building layouts; some tied to theoretical frameworks and others created from bottom-up heuristic approaches.

More broadly, methods for procedural generation of buildings and room layouts can also be found in fields of graphic design, computer-generated animation and video game development. The methods employed are diverse, but they typically involve techniques more tailored towards a specific application. They also tend to be most interested in generating "plausible" solutions that look "realistic" to a human viewer. With this goal in mind, some developers in this area have begun experimenting with deep learning algorithms.

Existing Work

A review of space planning methods by Veloso (2017) distinguishes between different types of methods, including: those that use “floating bubbles” that interact in pseudo-physical ways (applying a variety of heuristics, balancing of forces, measures of optimality, and methods that simulation artificial life, often applying evolutionary algorithms.

One study by Koenig (2012) approaches the space allocation problem by translating the topological relationship between layout elements into functional objectives in an optimization problem, applying genetic algorithms for generative element layouts, and using a connection tree to maintain circulation requirements. Koenig also considered application of dense packing theory, accounting for a form of collision detection and applying slicing trees to subdivide the problem (2014).

Another study applied a two stage algorithm, with a multi-agent topology finding system generating possible configurations of rectangular spaces in 3-dimensions, and an evolutionary optimization process to refine the solution (Guo, 2017).

Of the first type, one study applies perturbation through mutation to support convergence, relying on a memory of past solutions to avoid sub-optimal local optima (Hua, 2010). Other areas of investigation include application of Markov chains to retain “stories” related to building program (Christensen, 2014), exploration of interactive elements (Bayraktar 2013), incorporating architectural space syntax (R Schaffranek 2013).

Layouts and Energy – Preliminary Context Study

An illustrative case study was devised to demonstrate the impact that layout decisions can have on the energy performance of a building, independent of other factors.

The baseline was constructed in EnergyPlus as a square (576 m^2) co-working facility located in Toronto, with evenly distributed 50% window-to-wall coverage and NECB 2015 enclosure performance. The building is divided into 9 equal zones (along a grid), 5 of which are typical open office, 3 are shared multipurpose/event spaces and 1 is a server room – each with corresponding space loads, schedules and setpoints. Dividing the spaces are semi-modular partitions that allow 0.2 air changes per hour to exchange.

Without changing any other contextual, building program or design parameters, the complete set of 504 layout permutations (space-types assigned to the 9 zone grid) were simulated for a typical meteorological year.

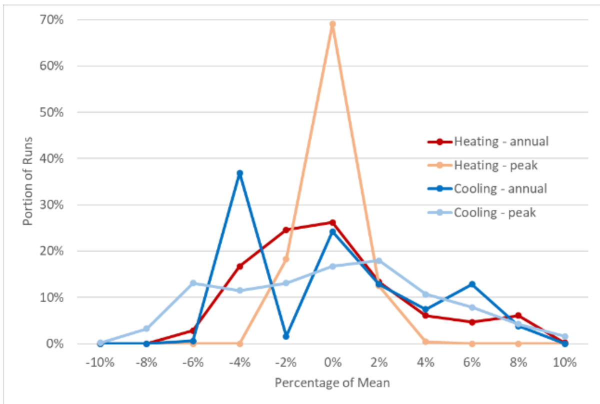

Figure 1: Thermal energy demand results for 504 runs under varying layouts but equivalent building program

Figure 1 shows the results for annual thermal heating and cooling aggregate demands and peak loads. The peak cooling results vary over a range 19.9% of the mean (86 W/m^2), with annual heating and cooling varying by 16.7% and 12.9% of their respective means (117 and 42 kWh/m^2), and peak heating varying by 5.5% relative to its mean (100 W/m^2).

These values are significant in the context of thermal energy performance and design decision-making, demonstrating the importance of layout (even without accounting for variability in massing and built form).

Other considerations that would further compound the impact of layout decisions on energy (and other performance objectives), and vastly increase the problem space include:

- Variations in building geometry
- Passive airflow networks and indoor air quality
- Active transfer air systems and shared plenums
- System boundaries; duct and pipe networks
- Active heat recovery and load sharing
- Passive elements and thermal storage
- Occupant comfort
- Constructability and staging

The remainder of this study will focus on the core problem and generic “adjacency” priorities, which can ultimately stand-in for many of these aspects of building performance simulation.

Room Topology – Graph Representations

A number of visual tools can be used to help understand the room data, including cord diagrams, spanning trees, and networks.

Many different fields (from highly theoretical to very practical) have incorporated elements of graph theory when addressing the problem of arranging physical objects in Euclidean space (though not always with explicit intent).

The building program can be readily represented by a weighted, undirected network of nodes (spaces) and edges (adjacency information). Drawing a network diagram is often an exercise in aesthetics, and there are a number of heuristic methods used in practice. Some more common methods in network theory include Fruchterman-Reingold and Kamada–Kawai algorithms, which incorporate aspects of physics modelling to iteratively refine the placement of the nodes and edges; however, most of these aim for "good enough" visual clarity and organization, without explicit intent to "optimize".

At a higher level, there are concepts in Graph Theory for representing graphs as circles with variable area (e.g. contact graphs). The Circle Packing Theorem shows how a graph can be represented by the points of contact between densely-packed, non-overlapping circles of variable diameter.

Most of the methods described above do not explicitly attribute physical dimensions to the nodes and edges (i.e. nodes are a represented, even graphically, as single points; and the length of the edge is irrelevant). While room area and physical distances could be encoded into more complex network structures, this abstraction would mean that interpretability would be diminished.

Outside of graph theory, this is somewhat analogous to the less abstract but more general class of "dense packing" problems (minimizing or eliminating space between objects of various sizes and physical characteristics), which in turn may draw on physics simulators.

Realizing this analogy, the conceptual framework used in this study combines some of the more abstract network theory concepts with a more explicit representation of the problem (building spaces with physical dimensions and connections). This allows useful information for the specific application to be retained in the model (room shapes, sizes and positions in Euclidean space), and used to guide optimization in a direction that is both more efficient and more intuitive to human users.

In its simplest form, the prioritized room adjacency information can be encoded into a matrix of values representing the weighted connection between each room, and applied to the edges of the graph. In the same way, additional parameters could be included to capture a broader set of objectives (such as thermal zoning, occupancy schedules, HVAC system distribution, etc.).

Methodology

The overall optimization process is divided into three main stages for implementation:

1. Room topology - initial placement of rooms
   - Connection trees, adjacency matrices and architectural graphs
   - Implements a force-directed algorithm (rooms as pseudo-physical bodies)
2. Space allocation - full layout generation
   - Implements a cellular-growth algorithm (rooms as clusters of cells)
3. Heuristic refinement - layout improvements and plan detailing (Future Work)

Discussion of potential improvements, current limitations and practical considerations are included with the methods below.

Space Data

Defines the room areas and weighted adjacency matrix for *n* spaces. The methods will accept any number and size for rooms – provided everything is entered using consistent units – but for the purposes of this study, a case study of 10 rooms with random sizes and adjacency weightings were created.

Additional matrices for other relationships between spaces can be easily added in similar ways, to be accounted for in the objective function calculations and optimization, including:

- thermal zoning
- occupancy schedules
- HVAC system distribution

Three other vital design elements are not yet included in this version of the method:

1. Egress/circulation: The connections between spaces and paths to exit are important for Building Code, accessibility and programmatic considerations. These could be incorporated using additional room elements (nodes) with special features (such as a non-circular shape), additional features for each room, or handled in a later stage of the overall process.
2. Exterior adjacency requirements: Ensuring some rooms are places along the perimeter is important in practical layout design. This could be handled using additional "room" elements (many "outdoor" nodes that are either static or dynamically repelled outwards – each room would have a special companion outdoor node that must be adjacent, of a size sufficient for at least a courtyard, and attracts to the orphan outdoor node cluster), an overlay of another force-vector field (similar to a global "anti-gravity" well, but pushing outwards), or some other parameter embedded in the room definitions themselves. This aspect could also be handled in a later stage of optimization.
3. Site boundary/perimeter constraints

Visualize the relationships between rooms

The weighted adjacency matrix is shown in Figure 2 as a heat map, with the colour scale representing the strength of the bonds between rooms (as defined by the initial problem set up).

Other visual representations are possible and could be implemented here to help visualize the problem (i.e. network plots, graph representations, cord diagrams, etc.).

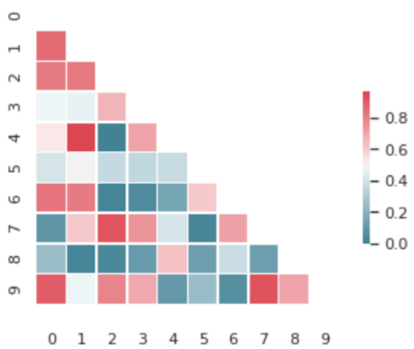

Figure 2: adjacency weights between case study rooms

Stage I Method - Graph Layout and Agent-based Optimization

The objective being minimized in this method is the "potential energy" of the system, defined by the agents (rooms/nodes), their positions (in Euclidean space) and a set of forces (based on adjacency priorities, physical constraints, and other parameters).

This method converts the weighted, undirected, complete graph data for nodes (rooms) and edges (adjacencies) into pseudo-physical system representation in Euclidean space (room areas and locations). Initial positions for the rooms are shown in Figure 3.

Circles were chosen for preliminary representations of the rooms to preserve physical space for each room, while enabling more robust optimization methods drawn from graph theory, the next stage of optimization involves determining the shapes and delineations between spaces. Therefore, only the position information will be passed to the next stage of optimization.

The "forces" acting on these agents are analogous to those in basic physics simulators, with magnitude calculated based on entered parameters and system state:

- gravitational: attraction to a global reference point and/or the centroids of other agent "masses"
- elastic: force that grows in magnitude based on distance from reference (in this case, minimum force occurs when rooms are in contact, but not overlapping)
- strong repulsive: a force defined as proportional to the amount of overlap area between rooms (i.e. zero force when rooms are not in contact), which can be increased to aid convergence on dense-packing solution (shifting from "soft-body" to "hard-body" representation)

Based on the current state of the system, the optimization method calculates the cumulative force (net magnitude and direction in Euclidean space) acting on each agent (room) and uses this to determine how "far" the agent will move over the next time step.

After this is completed for each agent, a new system state is determined. This quasi-steady-state approach iteratively refines the solution to minimize the "potential energy" of the system.

The first sub-method included (*minimize()*) is based on a simple heuristic approach, which iteratively moves the circles together. The primary optimization algorithm for this stage is *optimize()*.The algorithm includes a variety of parameters that can be adjusted to control the optimization process. The method could be improved using a separate algorithm to select "good" initial states based on system information, such as the Kamada–Kawai algorithm.

There are a number of other areas the algorithm itself could be improved, and that might be revealed through further investigation. For example, the inclusion of some randomness might aid convergence; specifically, random "perturbations" (similar to "shaking" a physical system) might help the system avoid local optima and settle on a global optimum. Along with a more organized encoding of the parameters, this would also support the use of genetic algorithms to more efficiently search the problem space. Another area would incorporating "memory" of past solutions (similar to a Markov Chain or control theory) to facilitate convergence.

This method has been set up with a number of parameterized components that would make it straightforward to plug into more advanced machine learning algorithms for exploration of a broader problem space. This would be especially useful if multiple, competing objectives were incorporated (such as minimizing thermal demand versus preference for perimeter exposure).

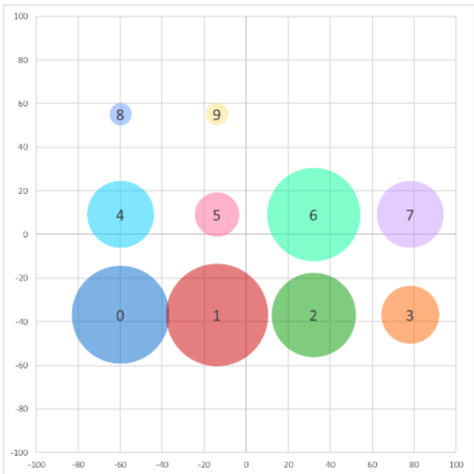

Figure 3: Initial case study room areas

Stage I Results

The solution state is comprised of a set of optimal positions for the rooms based on the objective of minimizing attractor and repulsor forces, under constraints preventing overlap, assuming the rough "circle" representations. The room layout class can initialized with a variety of parameters controlling the various forcing functions, constraints, objectives, and heuristic components.

For this study, the agent-based optimization methods were applied to identify (potentially) optimal solutions for four illustrative sets of parameters. The results over the full set of iterations are shown in Figure 5.

As a frame of reference, the final "potential energy" for the basic heuristic solution (*minimize()*) was 1505. It can be seen that the method produces better results than the reference heuristic (with system "potential energy" between 1052 and 1160), but that the rate of convergence and final solution vary depending on the parameters used. A much more rigorous parametric sensitivity analysis should be completed as part of future work.

Preliminary Space Allocation

As described above, a review of literature and practice reveals a wide variety of methods that have attempted to procedurally generate architectural layouts.

This study employs a multi-stage optimization approach, with Stage II designed to transform the optimal positional solutions identified in Stage I into actual building layouts. This involves placing the positional information for each space in the context of the whole site, and determining optimal room shapes and delineations.

Illustrative Visualization - Voronoi Diagrams

Drawing on clustering algorithms is one potential approach for forming 2-D shapes from the 1-D positional solution (provided in Stage I).

As an illustrative example, the Stage I optimal solution can be used to generate a Voronoi Diagram (Figure 4). The ridges that form the edges of the polygons are defined by being equidistant between nearest points.

External points were added for the purposes of aiding visualization - further refinement of this approach is not explored in this study, as a different direction was taken for the Stage II optimization.

Some research has shown promise using the sub-division approach, and a possible extension of this Voronoi exercise would be to create a custom clustering algorithm that employs weighted distance calculations (e.g. Chebyshev, which helps preserve rectilinear shape) to generate a more realistic initial solution that could then be refined.

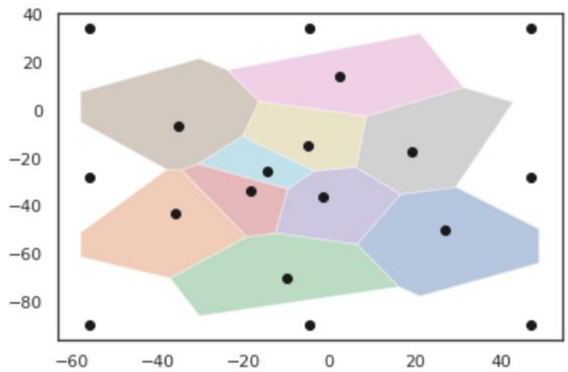

Figure 4: Voronoi diagram from the Stage I solution

Stage II Method - Agent-Based Gradient Ascent and Cellular Growth

A different agent-based optimization strategy is employed for Stage II. The optimal locations identified in Stage I are used as "seeds" planted in an overall grid representing the building site (sub-divided using the same units as previous stages). From these starting positions, each agent (room) grows outwards a unit cell at a time, ultimately forming a cluster equal to the total area of the room.

This cellular growth is driven by selection of the most promising new cell from adjacent options, and is in turn constrained by its own size, the site boundaries and presence of other cells. Each agent is encoded with rules for growth which can also respond to environmental factors (like following a "nutrient gradient"). Specifically, the following have been implemented in this method:

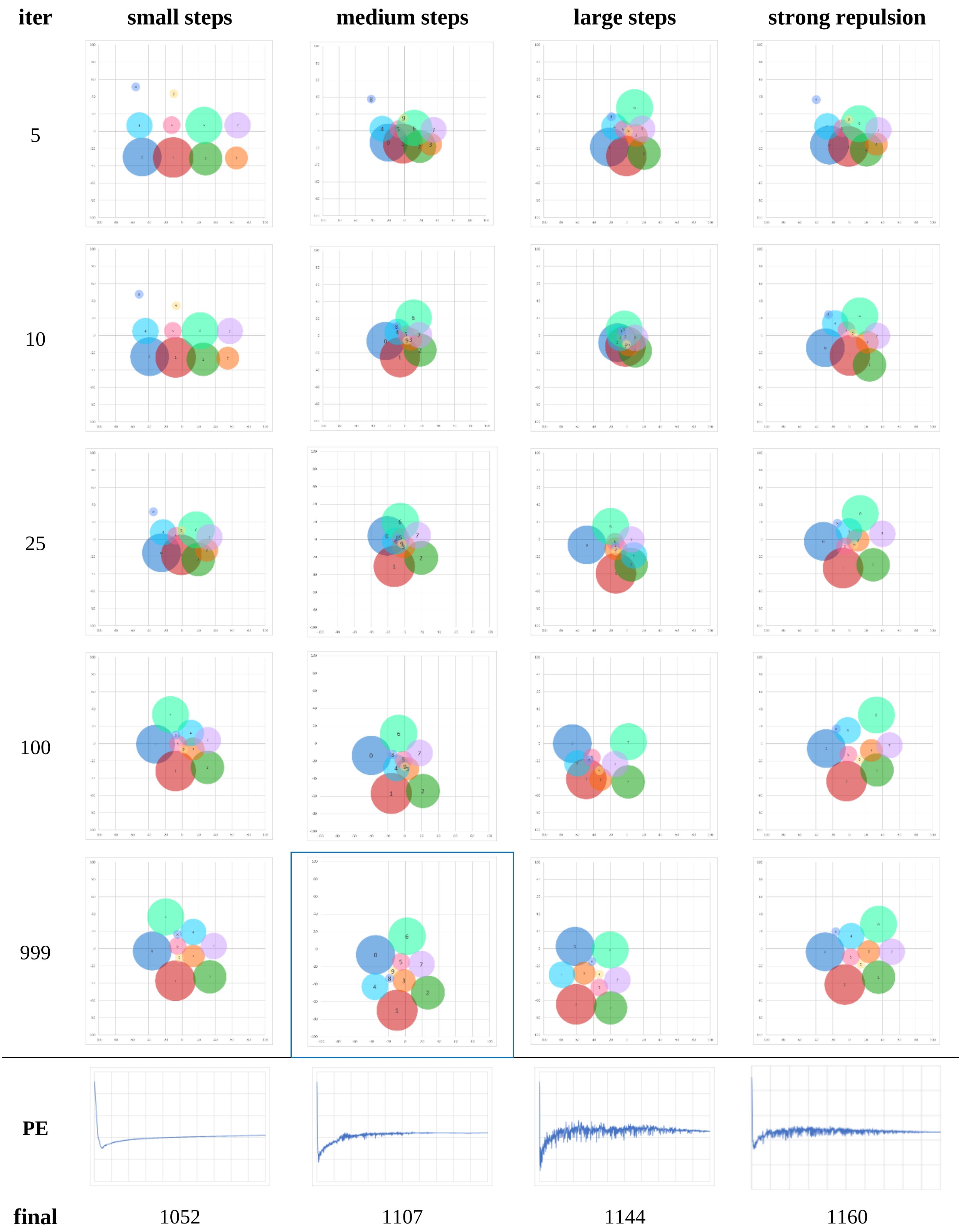

Figure 5: Stage I agent-based graph optimization for a case study building layout using four distinct sets of parameter assumptions, with the selected 'optimal' solution highlighted.

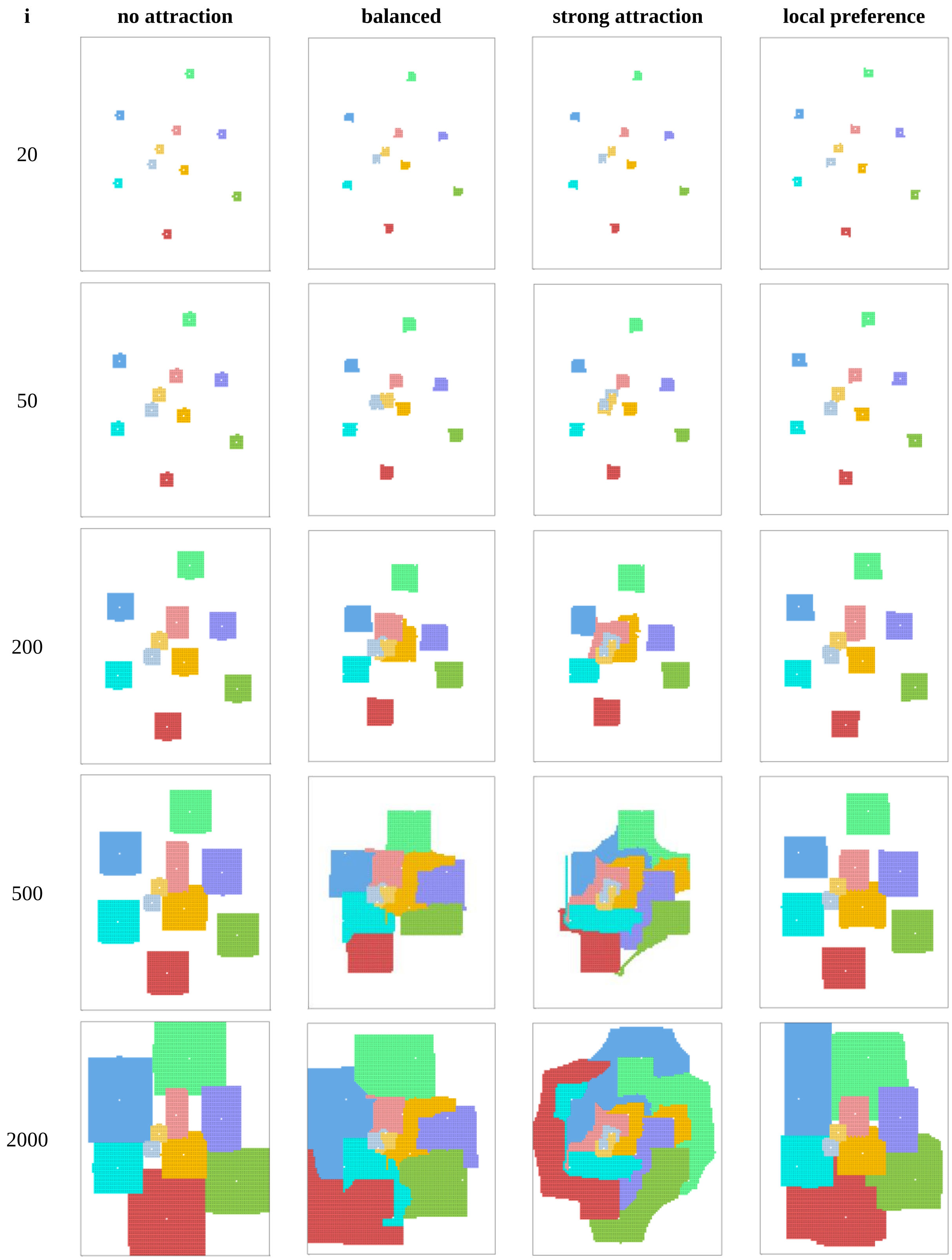

Figure 6: Stage II agent-based cellular growth optimization for a case study building layout using Stage I results as initial conditions but four distinct sets of parameter assumptions

- maintaining an aspect ratio in line with a pre-established target
  - i.e. prioritizing directions that do not deviate from the target
- maintaining spatial density
  - i.e. prioritizing cellular growth that "fills out" a generally rectangular form
- preference for cells local to original centroid
  - i.e prioritizing cells that are nearer the original seed position
- climbing attractor field gradients
  - i.e prioritizing cells that are nearer other rooms with high adjacency weighting

These are parameterized and packaged in an evolving fitness function, with agents attempting to maximize their own fitness through cellular growth at each iteration.

Similar to Stage I, once multiple objectives are included, additional randomness combined with and algorithm better suited for robust global search (e.g. genetic algorithms).

The attractor field should be dynamic, evolving as the clusters grow to capture changes in distance. It could also be represented as a matrix of vectors (rather than ranks), capturing direction and magnitude, and perhaps better suited for more linear algebra techniques that might be useful with more complex multi-objective problems.

Implementation

This sequential method keeps track of the growth of each room (cellular cluster i) in the context of the site, as well as the attractor field matrices for each room (i), which need to take into account both proximity (to each other room j) and relative weight of adjacency (i,j).

This method includes the initial placement of the room "seeds" based on the Stage I solution, along with creation of instances of the main Room() class for each agent, followed by iterative calls of the growth function contained in the Room() objects until each has satisfied their target area.

A record of the iterative growth is stored in a 2-D dataframe, which can be output to another csv for external processing.

Stage II Results

Preliminary results from the Stage II optimization method are shown in Figure 6 for four different sets of method parameters (using the "balanced" solution set of initial positions from Stage I as seeds).

It was apparent that greater attractive forces disrupted the default tendency for rooms to maintain their aspect ratio centred on their initial seed. A balance between the forces (through trial and error) yielded the best result, Furthermore, the local preference forcing function was vital to keep the initial seed position near the final centroid position.

This method would be improved with the inclusion of a global objective (similar to the "potential energy" from Stage I). It would need to be consistent across feasible solutions to allow comparison and refinement.

Quantification of performance and more rigorous parameter tuning should be part of future work.

Stage III Method - Heuristic refinement

As can be seen in the final solutions from Stage II, layout improvements and plan detailing are still required. Some major elements could be included in the earlier Stages (rough representations or corridors and preference for exterior exposures), but detailed refinement of spaces could be.

Possible methods include establishing a set of heuristics for "swapping" of cells, perhaps involving a gradient ascent towards better performance under a more refined objective function; one that takes into account both individual agent fitness and constraints, as well as global targets and requirements.

There is also potential to incorporate qualitative criteria in this process, and to generate interactive tools that would allow designers or researchers to adjust and refine the potential building layouts.

Conclusions

The multi-stage framework presented in this study shows potential as a flexible aid to researchers and designers in their exploration of the building layout problem space.

To bring the method to a level where it can be applied as a tool in practice, the following additions would be necessary:

- Sensitivity analysis and parameter tuning for Stages I and II
- Identification and incorporation of missing key features (circulation, exterior, etc.)
- Development of Stage III optimization and refinement
- Re-factoring the code into a cohesive object-oriented framework, along with creation of some minimal user interface

Promising directions that research and development of this method could take include:

- Addition of multiple objectives (thermal, mechanical, occupancy-based, etc.)
- Connection to thermal loads and building energy simulation
- Employment of qualitative research methods to determine the best fit for such a tool in practice, and perhaps to add a layer of qualitative criteria for evaluation of feasible solutions
- Deeper investigation and refinement of room data representations, constraints and forcing functions, heuristic rules
- Incorporating additional parameterization and more randomness to support the use of evolutionary algorithms or other meta-heuristics in exploration of a much broader problem space
- Further review of network theory and other related fields of study for useful concepts or methods, and to help form a more rigorous conceptual framework

References

Bayraktar, M. E., & Çagdas, G. (2013). Fuzzy Layout Planner: A simple layout planning tool for early stages of design. *Spatial Performance and Space Syntax - Volume 1 - Computation and Performance - eCAADe 31*, 375-382

Christensen, J. T. (2014). The generation of possible space layouts. *Proceedings of the 32nd eCAADe conference*. 239–246

Guo, Z., & Li, B. (2017). Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system. *Frontiers of Architectural Research, 6(1)*, 53–62.

Hua, H., & Jia, T.-L. (2010). Floating Bubbles An agent-based system for layout planning. *New Frontiers: Proceedings of the 15th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA*, 175–183

Koenig, R., & Knecht, K. (2014). Comparing two evolutionary algorithm based methods for layout generation: Dense packing versus subdivision. *AI EDAM, 28(3)*, 285–299.

Koenig, R., & Schneider, S. (2012). Hierarchical structuring of layout problems in an interactive evolutionary layout system. *AI EDAM, 26(2)*, 129–142.

Richard Schaffranek, & Michael Vasku. (2013). Space Syntax for Generative Design: On the application of a new tool. *Ninth International Space Syntax Symposium, Seoul.*

Veloso, P., Rhee, J., & Krishnamurti, R. (2017). Multi-agent Space Planning. *CAAD Futures 19*, 130-152.