Beyond Heuristics

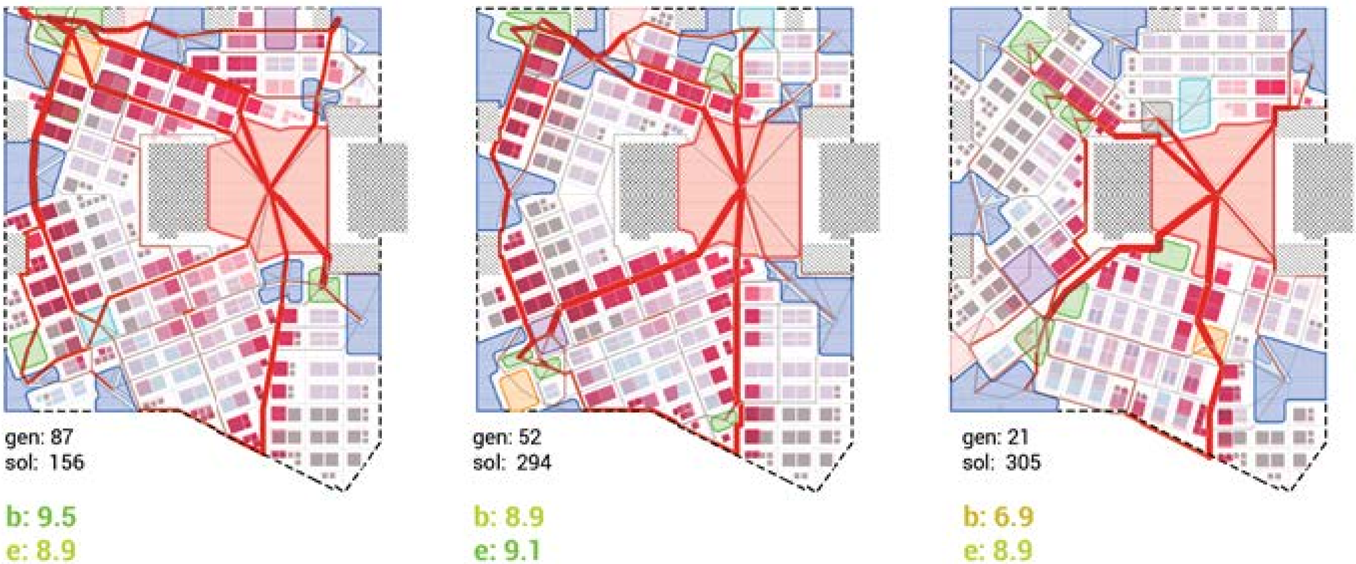
Danil Nagy
The Living, an Autodesk Studio

Lorenzo Villaggi
The Living, an Autodesk Studio

Dale Zhao
The Living, an Autodesk Studio

David Benjamin
The Living, an Autodesk Studio

A Novel Design Space Model for Generative Space Planning in Architecture



1

1 Three high-performing designs from space plan optimization.

ABSTRACT

This paper proposes a novel design space model that can be used in applications of generative space planning in architecture. The model is based on a novel data structure that allows fast subdivision and merge operations on planar regions in a floor plan. It is controlled by a relatively small set of input parameters and evaluated for performance using a set of congestion metrics, which allows it to be optimized by a metaheuristic such as a genetic algorithm (GA). The paper also presents a set of guidelines and methods for analyzing and visualizing the quality of the model through low-resolution sampling of the design space. The model and analysis methods are demonstrated through an application in the design of an exhibit hall layout. The paper concludes by speculating on the potential of such models to disrupt the architectural profession by allowing designers to break free of common "heuristics" or rules of thumb and explore a wider range of design options than would be possible using traditional methods.

INTRODUCTION

Architectural design is an extremely complex task. A typical design project may take several years to complete, with many interrelated decisions that must be made along the way. Since human designers are limited in their ability to consider all these decisions at once, we tend to break down complex problems into a sequence of smaller problems that can be more easily solved based on "rules of thumb," or strategies that have been proven to work in the past. In computer science, simple proven strategies for solving complex problems are called "heuristics." Although heuristics can be useful for efficiently generating workable solutions, when applied to complex problems, they are not guaranteed to produce the best overall solution. In the context of design, they can also lead to "design fixation," where "designers limit their creative output because of an overreliance on features of preexisting designs" (Youmans and Arciszewski, 2014).

To address the limitations of heuristic methods, the field of computer science has developed a set of optimization methods known as "metaheuristics." In general, metaheuristics are stochastic algorithms that can find optimal solutions to complex problems by iteratively sampling possible solutions, evaluating them based on specified performance factors, and using this information to derive better and better designs. The application of metaheuristic optimization algorithms to architectural design problems is commonly referred to as generative design. With generative design, architects can leverage the power of computing to explore a much wider range of solutions than is possible with traditional design methods. Because the algorithms have no inherent intuition or bias for solving specific problems, they can help architects break free of the heuristics found in traditional design processes, leading to the discovery of not only high-performing design solutions but also novel ones.

The successful application of these methods relies heavily on the specification of a well-defined design space model. A design space model is a digital model that defines a set of design solutions based on a finite set of input parameters, and automatically evaluates each solution based on one or more pre-defined metrics. This model forms a conceptual "space" containing all possible design options, with each input parameter defining a dimension of the design space. During optimization, a metaheuristic algorithm "searches" this space by stochastically sampling design options, evaluating their performance, and using this information to find better designs.

The formulation of the design space model is critical to the optimization process because it describes both the range of possible design solutions (scope), as well as how the designs are related within the space (structure). However, while the application of metaheuristic search algorithms to design problems has been well-explored in the literature, there has been relatively little study of how designers should go about designing such models, and how they can be analyzed and evaluated before the optimization is performed.

In this paper, we describe a novel design space model that can be used to solve two-dimensional space planning problems in architecture. Space planning is one of the most complex problems in architecture, and deals with the optimal arrangement of a set of programs or features within an architectural floorplan (Homayouni 2000). Furthermore, we present a set of guidelines and methods for evaluating the scope and structure of a design space model to decide whether the model is fit for optimization using metaheuristic algorithms. To demonstrate these methods, we show their application in the design of an exhibit hall layout.

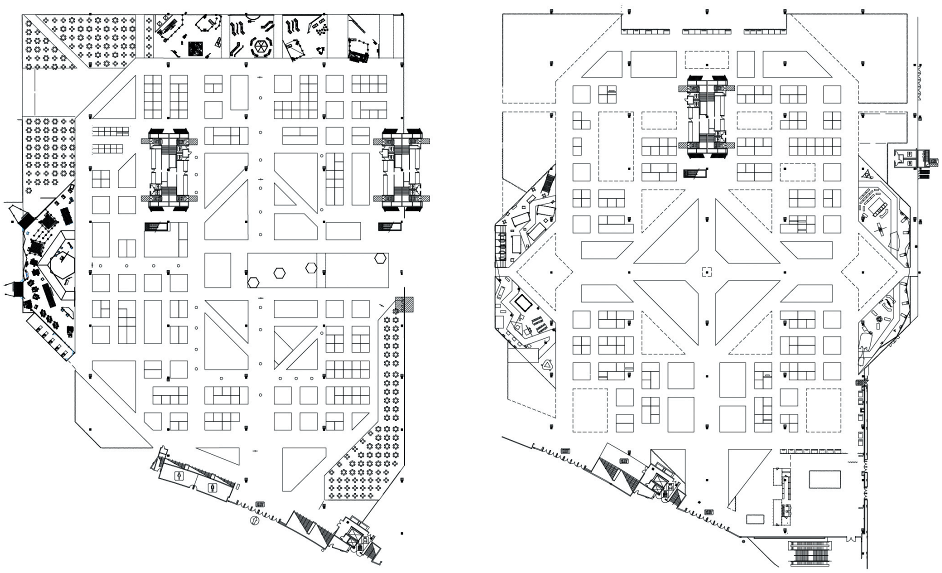
LITERATURE REVIEW

The application of metaheuristic optimization algorithms toward space planning problems has been widely explored, as evident in the excellent review by Calixto and Celani (2015). Among the research cited in this review, the most relevant to our design space formulation is the work of Flack and Ross (2011), who propose a space planning model based on the subdivision and merging of contiguous plan areas. Our work extends this method to non-orthogonal layouts and proposes a novel data structure that optimizes the subdivision and merging operations.

While the research cited in this review explores a variety of different model representations and optimization algorithms, there is little discussion of how the design space model can be evaluated by designers before the optimization is run. We extend this work by proposing a set of guidelines that can be used to analyze the quality of a design space model based on its scope (the bias vs. variance tradeoff) and its internal structure (the complexity vs. continuity tradeoff). We also introduce a method for visualizing these qualities based on low-resolution sampling of the design space.

THE DESIGN PROBLEM

To demonstrate the effectiveness of our design space model we applied it to a real-world design project: the layout of a 288,000 square foot exhibit hall for a major event happening later this year. Our clients in the project were the event organizers, the event production company, and the facility management company. The aspirations of the project were to create a novel event layout that accommodated the programmatic needs of the event while maximizing exposure between exhibitors and attendees.



2

2 Event layout from previous two years.

3 Design space model definition:
(a) boundaries and constraints
(b) subdivision based on avenues
(c) regular subdivision into cells
(d) program seed parameterization
(e) cell merging based on seeds
(f) standard booth placement

An initial analysis of layouts from previous versions of the same event (Figure 2) revealed the application of many heuristics or "rules of thumb," such as orthogonal layouts and clustering of food and beverage programs. While these rules of thumb created reasonable solutions, they did not have any fundamental connection to the aspirations of the project and thus artificially restricted the range of possible design solutions. By applying our design space model and optimization method, we were able to break free of some of these limitations, delivering a layout proposal that was unique yet high-performing based on the desires of the client.

THE DESIGN SPACE MODEL

Our design space model is based on the morphology of urban street layouts. Starting with the boundary of a two-dimensional space, the model uses a series of "avenues" to subdivide the space into a collection of smaller parcels. It then merges some of those parcels to form larger regions to accommodate the programmatic requirements of the layout. In the rest of this section we will describe how the design space is defined through a set of constants and boundary conditions, how it is parameterized to produce a variety of space plan options, and how each option is evaluated relative to the goals of the project.

Design Space Definition

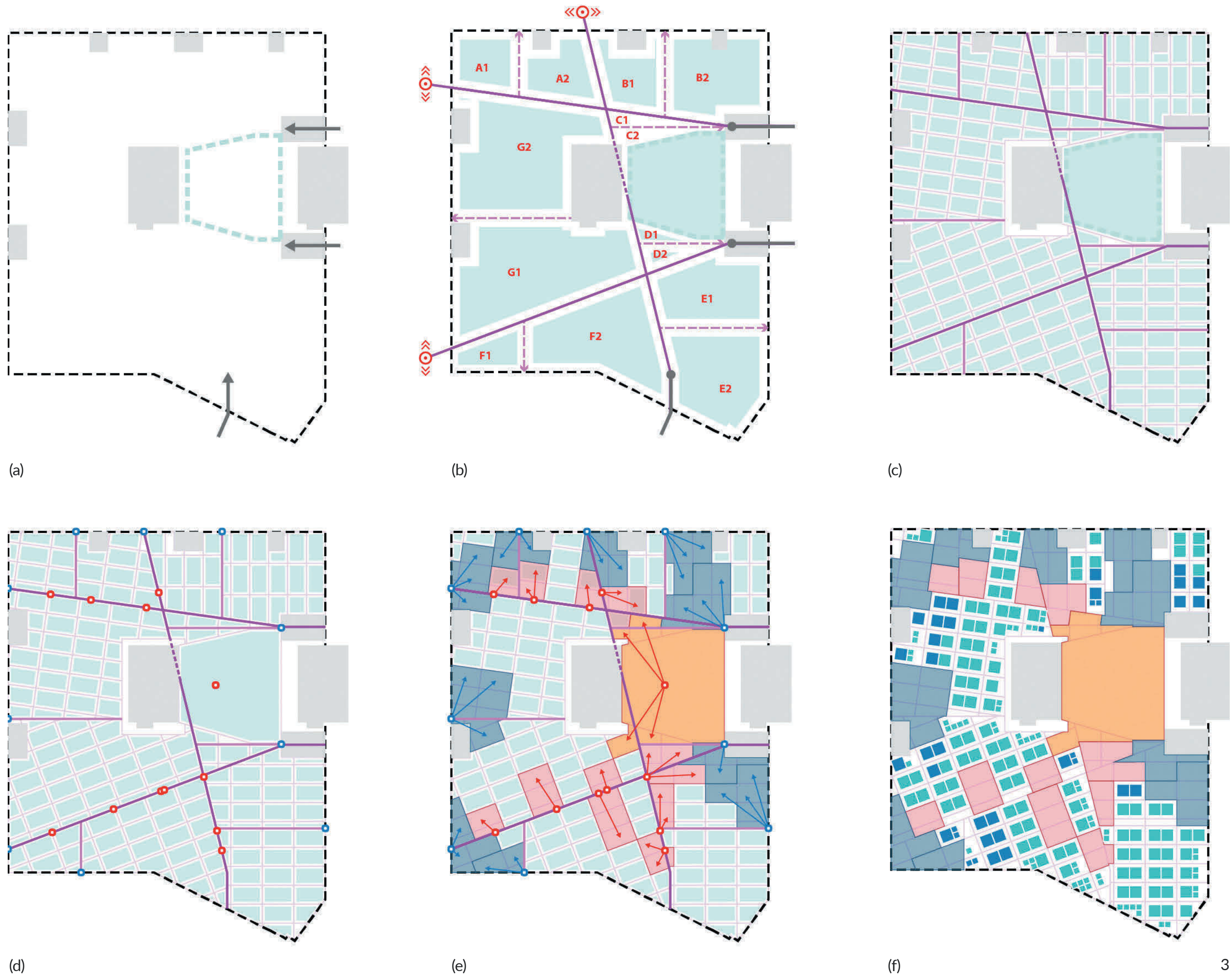
The boundary of the design space is determined by the walls of the exhibit hall and represented in the model as a closed polyline (Figure 3a). Another set of polylines define "no-go zones," such as egress areas, restrooms, and vertical cores where no event programs can be placed. A final set of constants are the three entrances to the exhibit hall. These entrances are represented by points and define the starting point of three avenues, which are the basis of the subdivision process described in the next section.

Design Space Parameterization

The design space is parameterized using two sets of parameters. The first set of three parameters guides the placement of three main circulation avenues that subdivide the plan into a set of smaller cells. The second set of 22 parameters guide the placement of the 11 major expo programs along the circulation routes, which then grow by merging adjacent cells until each program's size requirement is met. These circulation and program parameters describe a 25-dimensional design space that contains a wide range of valid plan layouts for the exhibit hall.

The first step of the subdivision process is the placement of three main avenues that connect the three expo hall entrances with a point on the opposite wall (Figure 3b). The location of this point is parameterized in the domain of the polyline edge representing the wall directly opposite each entrance. The avenues are used to subdivide the plan boundary into a variable number of macro-regions, which are then further subdivided into micro-regions according to the following procedure:

1. Identify macro-regions that share at least one edge with the exhibit hall boundary.
2. For each of these macro-regions identify the longest edge that is shared with the exhibit hall boundary and locate the midpoint of the edge.
3. Draw a secondary avenue line perpendicular from this midpoint to the opposite edge of the macro-region.

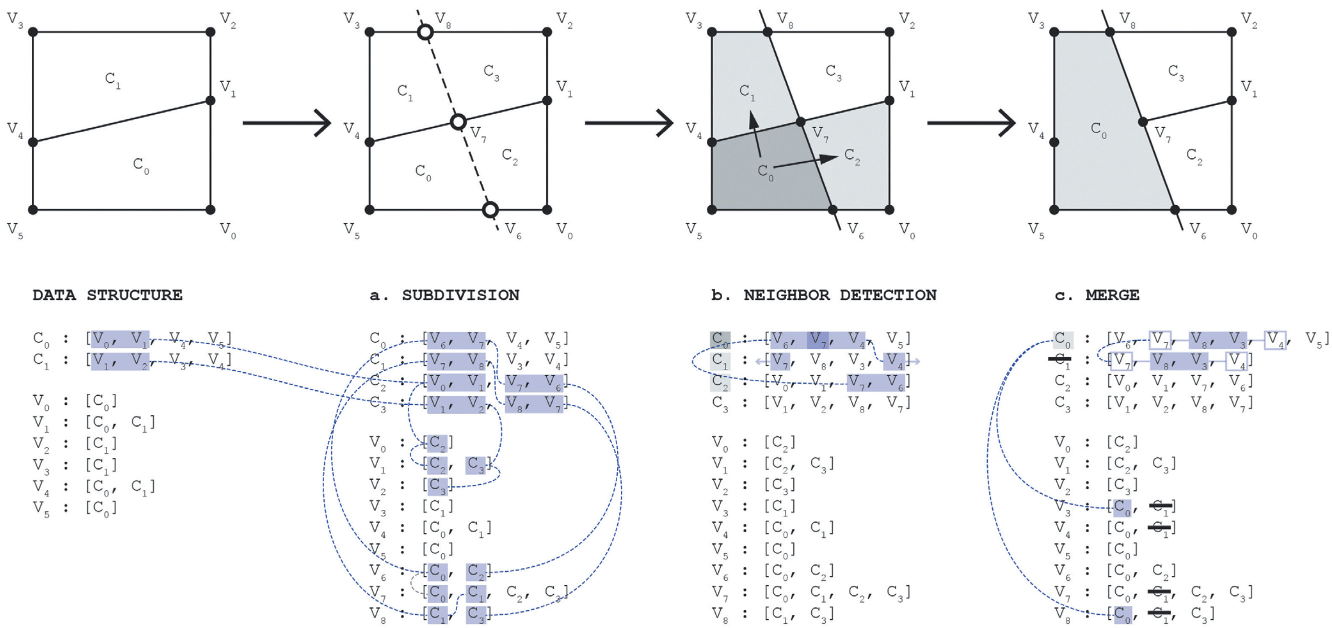


3

Finally, each micro-region is further subdivided using a fixed grid of 40 x 20 ft, which is aligned to the longest edge of the associated micro-region. This process results in a total subdivision of the original plan boundary into a series of smaller cells that are then used to place the various event programs (Figure 3c).

After the plan has been subdivided into smaller cells, the next step is the placement and allocation of space for the event's programs. Based on references from past events and early conversations with the clients, we generated a list of 11 major programs and associated area requirements. Following the city example, these major programs are placed along the three main avenues, forming "monuments" that serve as visual markers for visitors in navigating the space (Figure 3d). The placement of each of these programs is parameterized with a pair of parameters [j, k], where j is an integer in the range [0,2] that dictates the avenue along which the program is placed, and k is a float in the range [0,1] that dictates where it is located along the avenue.

In addition to the 11 main programs that are individually parameterized, there were an additional set of programs relating to the food and beverage (F&B) service of the event. To reduce the dimensionality of the design space, we chose not to individually parameterize the location of the F&B programs. Instead, these programs are always located at the ends of the primary and secondary avenues, where they hit the exterior walls of the expo hall. This strategy distributes the F&B programs around the floor plan, avoids dead ends, and minimizes congestion while creating attractions that draw people further into the space. Although they are not individually parameterized, the location of these programs is determined by the placement of the avenues, so they can be indirectly controlled by the optimization algorithm when testing various design options.



4 Cell mesh data structure and three primary operations.

Once the "seed points" of all the programs are placed, they are allocated in the plan according to the following procedure (Figure 3e):

1. For each program, the cell that contains the program's seed point is identified.
2. The neighbors of the starting cell are identified as the set of all cells that share at least one edge with the starting cell.
3. A neighbor is chosen for merging with the starting cell, such that the resulting area minimally meets the area requirements of the given program.
4. If all neighbors fall short of meeting the area requirement, the largest neighbor is chosen for merging. The process then repeats from step 2 until the area requirement is met.

The merging process results in a set of variable-size cells based on the needs of the program. Once the merging process is complete, the remaining cells are populated with a collection of standard 20′ x 20′, 20′ x 10′ and 10′ x 10′ exhibitor booths (Figure 3f).

Both subdivision and merging operations are implemented using a novel data structure (Figure 4) that allows fast subdivision and the union of a collection of planar contiguous cells with an arbitrary number of edges. The plan is represented by a collection of vertex and cell objects. Vertex objects have a physical [x, y] location in the plan, and store a list of pointers to cell objects to which they belong. Cell objects are defined by a list of vertex objects, ordered counterclockwise, which represent the polygonal boundary of each cell. Based on this data structure, we can implement efficient methods for neighbor search, subdivision, and merging without resorting to expensive geometric operations such as nearest neighbor searches, booleans or splits.

The dual steps of subdivision and merging create a complex design space capable of creating a very large variety of design options that are guaranteed to accommodate the programmatic needs of the event. While all potential design options are valid, some options are clearly better than others, and it is the goal of the optimization algorithm to search through the design space and discover the optimal designs. To do this, the algorithm must be able to evaluate each design based on one or more predetermined performance metrics—or goals—which are the focus of the next section.

Design Space Metrics

Based on discussions with the clients, we determined that the most critical goal of the event layout was the distribution of foot traffic, such that all exhibitor areas are evenly activated without creating undesirable congestion in any particular area. While a certain level of congestion is important for activating the various programs and exhibitor booths, experience from past events revealed that certain concentrations of high-activity programs such as F&B can lead to concentrations of congestion that are uncomfortable for attendees.

$$buzz = \sum_{i=0}^{n} routes\ length_i * w_i\ ,$$

$$w = amount\ of\ traversals\ in\ each\ route$$

$$n = number\ of\ nodes$$

$$exposure = \sum_{v\ \in G} length\left(edge_{v_i,v_j}\right),$$

$$v = vertices$$

$$G = graph\ of\ all\ sites\ with\ adjacency\ requirements$$

5 Visualization and calculation of output metrics buzz (left) and exposure (right).

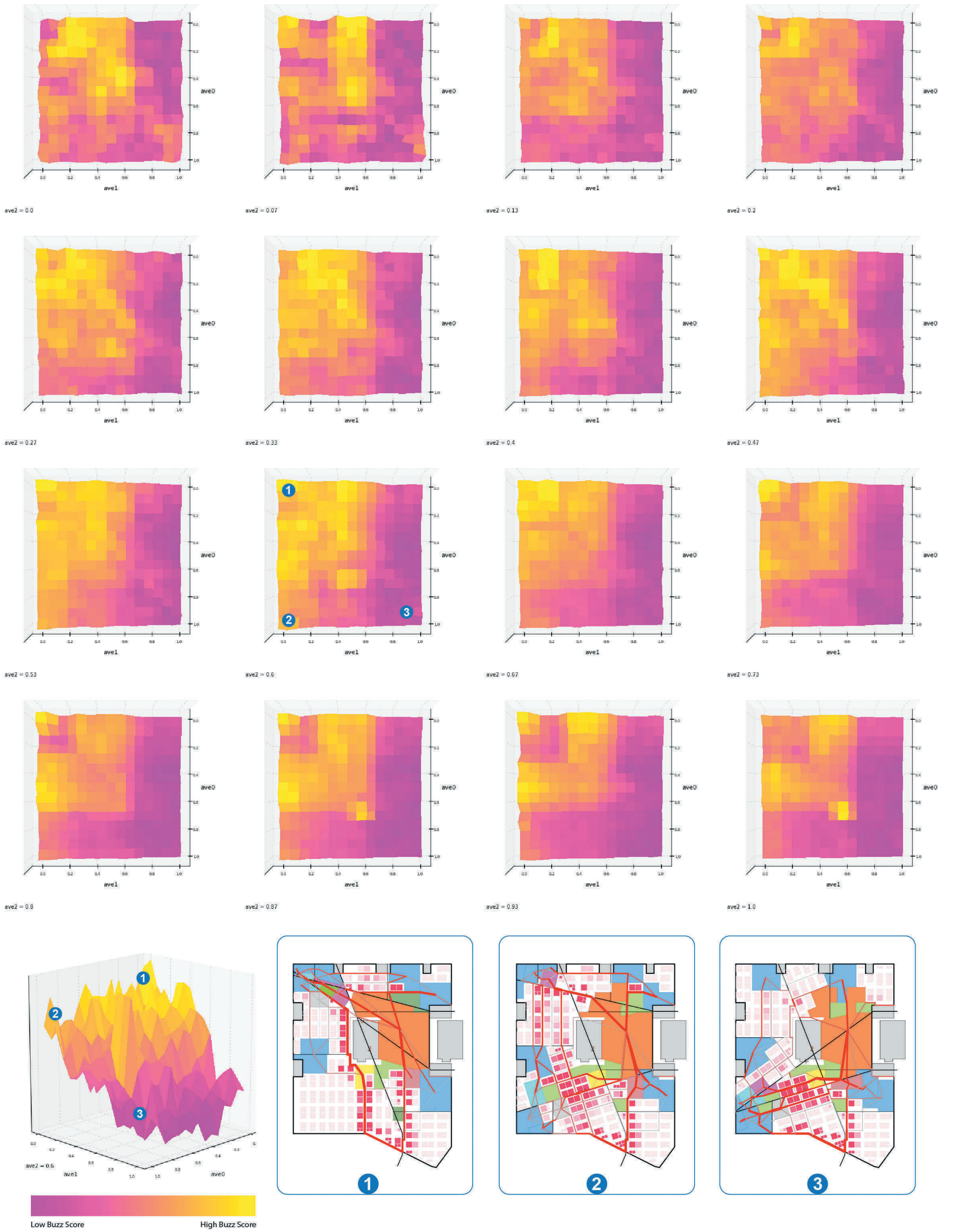
To capture these desires in the model we developed two related but distinct measures. The first one is called "buzz," and measures the spatial distribution of high traffic areas in the plan (Figure 4a). Higher values of buzz represent plans that not only create a large amount of foot traffic, but distribute this foot traffic evenly throughout the plan. The second metric is called "exposure," and measures the average foot traffic around each exhibitor booth (Figure 4b). Higher values of exposure represent plans where a large percentage of booths have a high level of traffic surrounding them. Both metrics were calculated using a novel static graph-based simulation method, which is fully described in a related paper.

The set of 25 design parameters and two metrics form the input and output interface of the design space model. Unlike some gradient-based methods, metaheuristic optimization algorithms do not rely on any knowledge of the internal workings of the model to derive optimal designs. Instead the model is treated as a generic "black box" function, which represents the complex mapping between the input parameters and the output metrics.

This is an advantage for design models, which are typically composed of many interrelated geometric operations, few of which can be analytically defined or differentiated. However, this also means that the mapping between the inputs and outputs of the model is a crucial factor in the success of the optimization. Despite this importance, there are few guidelines to help designers analyze the quality of their design spaces to ensure that the optimization yields good results. Without such guidelines, designers have no way of knowing whether the algorithm is efficiently searching the design space and finding the most optimal designs. In the next section we formulate two such guidelines, and demonstrate a visualization method for evaluating our model based on them.

DESIGN SPACE EVALUATION

To help designers evaluate the quality of their design space models we propose two guidelines formulated as tradeoffs between two extreme conditions of the model. The first tradeoff is bias vs. variance, which relates to the scope or breadth of the design space. A model with "high bias" is too simple, and does not capture enough variation in possible designs. You can think of such a model as being too "biased" towards a particular solution. On the other hand, a model with "high variance" is too flexible, and captures a much larger set of possibilities than is necessary for solving a problem. Such models often have a high percentage of invalid designs, which makes them difficult to search during optimization. A good design space model lands somewhere in



6 Design space evaluation.

between these two extremes—capable of creating a wide range of possible designs without being so vast that the search process becomes intractable.

The second tradeoff is complexity vs. continuity, which relates to the internal structure of the design space and the mapping between input parameters and output metrics. Complexity refers to the potential of the design space to generate unpredictable, non-intuitive results. A model with insufficient complexity is usually not worth the effort of optimization because we can easily determine the best design using our own intuition. Continuity refers to the internal consistency and structure of the design space. For the design space to be searchable, individual designs within the design space should maintain some internal relationships, which allows the search algorithm to navigate between adjacent designs and make valid predictions about the performance of designs based on the designs around them. Although these two properties are often at odds with each other, a designer's goal should be to maximize both—to create a model that is complex enough to allow unexpected solutions, yet continuous enough to be searchable.

To evaluate these tradeoffs in our design space model, we propose a visualization method based on a low-resolution regular sampling of the design space, which allows us to visually understand both the scope and the internal structure of the model. To perform the analysis we first narrowed our scope to the three avenue parameters since they have the most effect on the topology of the plan and thus the resulting value of the two metrics. We then created a design of experiments (DOE), which is a set of design options, based on a full permutation of a low-resolution sampling along the three parameters. In this case we chose to discretize each input into 16 steps, which yields 16 x 16 x 16 = 4,096 designs.

To smooth out the effect of the remaining 22 input parameters we also tested each design five times using random values for the other parameters and averaged the values of the output metrics. This gave us a total of 20,480 designs that had to be evaluated. The discretization of the main parameters and number of trials was fine-tuned based on the time it took to generate each design option and the total time we had to run the experiment. In this case, each design took roughly 20 seconds to compute, so the whole experiment took roughly five days to complete on a single computer.

Based on this sampling we can visualize the design space as a series of interpolated "response surfaces" with two input parameters mapped onto the x and y axes, and one of the metrics mapped onto the z axis. Figure 6 shows the resulting visualization for the "buzz" metric. Since we had to keep one of the input parameters constant in each plot, the visualization consists of 16 separate plots, with each plot representing a slice along the third input parameter.

Based on this visualization, we can draw some conclusions about the quality of our design space model based on the two tradeoffs. The plots show us that the transition between high and low scores is gradual in certain zones and more abrupt in others. In general, the landscapes show a degree of overall continuity while revealing local complexities between the input parameters and the output metric scores (Figure 6 inset). The design space also seems to be more continuous along the parameter represented along the y axis than the one along the x axis.

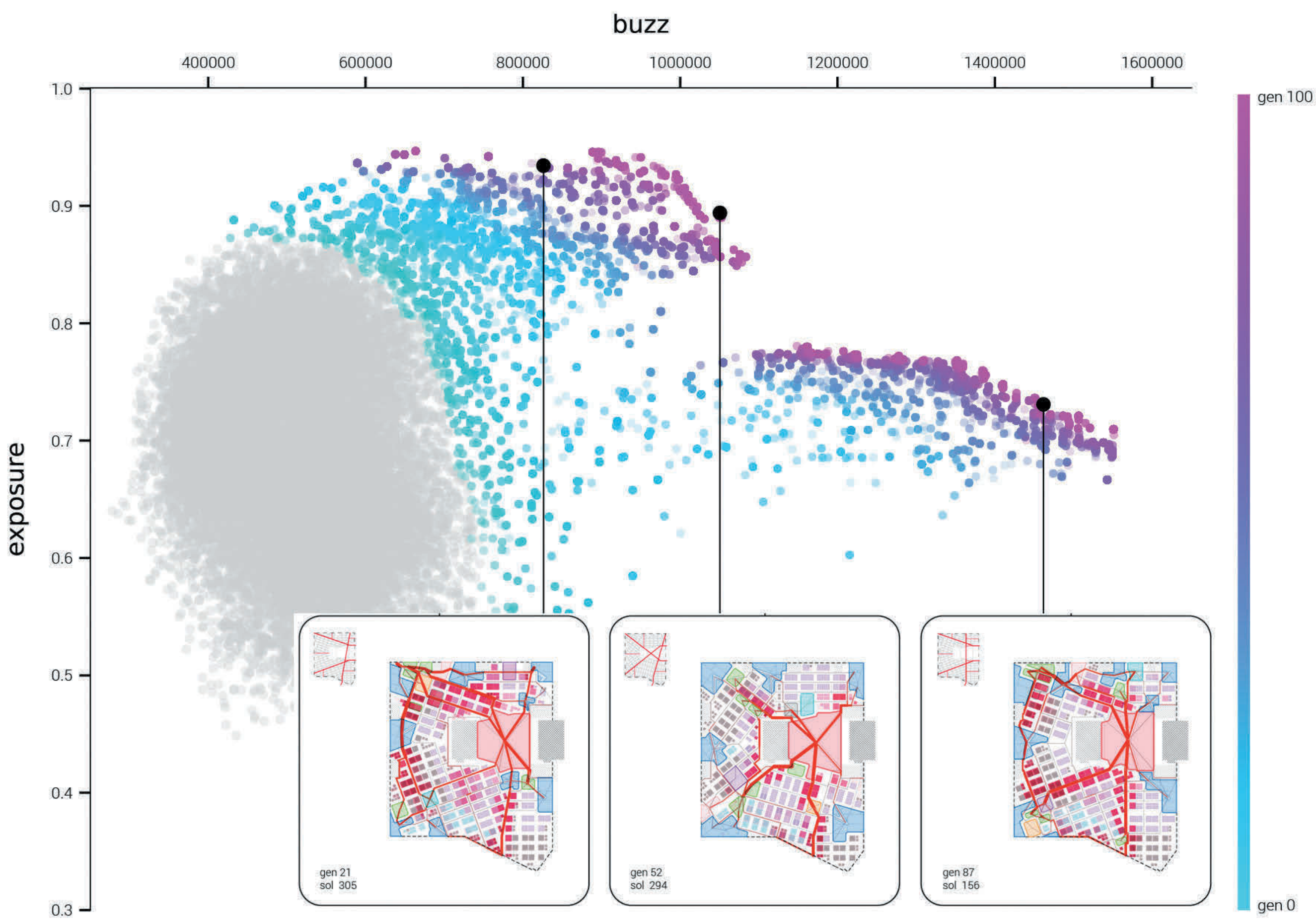
We can also learn about the scope of the design space by studying the form of resulting plans at its boundaries. As stated earlier, the merging process guarantees that all options within the space are valid designs. Thus we are not in danger of creating a model with too much variance. However, we can also see that the initial subdivision process leads to a wide variation of plan designs. Thus, our model is not overly biased towards particular solutions such as orthogonal layouts.

Finally, we can also study areas of high performance to deduce strategies that may lead to high-performing designs even before optimization. For example, we can see that the high performing region of the design space in the top left x-y quadrants is produced whenever both avenues miss the central obstacle in the exhibit hall. Now that we have located some potential strategies and have confirmed the relative continuity of our design space, we can be confident that the optimization process can search the design space and exploit those strategies to find the overall optimal designs.

DESIGN OPTIMIZATION

The final step of our design process was to run the design space model through a metaheuristic optimization algorithm to find the optimal designs. In this case we chose to use a genetic algorithm (GA), which despite being one of the oldest metaheuristic algorithms (Holland, 1975), remains one of the most popular and widely used (Marler and Arora 2004).

As indicated by their name, GAs find optimal solutions to complex problems by mimicking the evolutionary process in nature. This process begins by creating an initial generation of random solutions. These solutions are then ranked according to performance in the specified objectives. The best solutions are selected for "cross-over," a process in which pairs of solutions mix their input parameters to create a new generation of "children"



7 Results of optimization by objectives. Grey color represents designs from analysis stage, optimization designs range from cyan (early) to magenta (later).

solutions. Often, "mutation" is introduced by randomly altering the value of a small number of input parameters. By iteratively repeating this process over many generations of solutions, the algorithm can navigate to high-performing areas of the design space, eventually finding designs that are optimal, or close to optimal.

We performed the optimization using a variant of the NSGA-II genetic algorithm (Deb et al. 2002) with the following settings:

- Designs per generation: 320
- Number of generations: 100
- Avenue parameters mutation rate: 0.5
- Program seed parameters mutation rate: 0.3
- Cross-over rate: 0.9

One side effect of the visualization process described in the previous section is that we already had a large dataset of designs to start with. So instead of a typical first generation of random samples, we started the optimization with the 320 highest performing designs from our dataset. In addition to giving the optimization an early boost, this technique also reduces the chances of getting stuck in a local minimum because the initial designs were evenly sampled from the entire design space.

Figure 7 shows all the designs explored during both the analysis and optimization process plotted according to the two goals of the optimization (maximizing the values of both output metrics). The 20,480 designs produced during the analysis step are shown in gray, and the 32,000 designs explored during optimization range from cyan (early designs) to magenta (later designs). The plot shows that the optimization process is able to improve on the performance of even the most optimal designs found during the original low-resolution sampling of the design space. This demonstrates the utility of our hybrid process, which combines an initial low-resolution sampling of the design space to interpret the quality of the design space and produce an initial population of high-performing designs, with an optimization process that can further explore the design space to discover even better designs.

RESULTS

The optimization process produced a subset of 38 Pareto-dominant designs, which were all equally high-performing based on tradeoffs between the two metrics. From these we selected three designs that represented three distinct strategies for the event layout (Figure 1). These designs were then discussed with the clients, and the appealing features of each one were synthesized into a floor plan that was handed off to the production management company for further refinement.

CONCLUSION

Recently many architects and designers have become infatuated with the promise of generative design. Through the application of metaheuristic optimization algorithms, these methods can transcend the limitations of our traditional heuristic-based design processes, and help us discover novel and high-performing solutions to our most complex design problems.

As generative design tools gain wider adoption, the designer's role will be transformed from designing static three-dimensional objects to designing highly multi-dimensional design spaces that can be explored and optimized by artificially intelligent systems such as genetic algorithms. However, as designers begin to adopt these tools, they will need clear theories and guidelines for how to design a good design space model, evaluate its quality, and predict its ability to be optimized by an autonomous system. In other words, designing a generative design model requires experience, creativity, analytical thinking, and even "typologies"—just as traditional design does.

Although there have been many technical descriptions of the tools and some promising applications, we are still lacking a unified theory of design space design, and its wider applications towards the architectural design process. To address this shortfall, this paper provides three main contributions. The first is a description of a novel design space model for application in architectural space planning. The second is a set of guidelines for evaluating the qualities of a design space based on the bias vs. variance and complexity vs. continuity tradeoffs. The third is a set of methods for evaluating these tradeoffs through low-dimensional sampling and visualization. It is our hope that a deeper focus on the design implications of these technologies will lead to the adoption of these tools by a wider set of designers, causing a disruption in the architecture industry towards more innovative and higher performing designs.

REFERENCES

Calixto, Victor, and Gabriela Celani. 2015. "A Literature Review for Space Planning Optimization Using an Evolutionary Algorithm Approach: 1992-2014." *Blucher Design Proceedings* 2 (3): 662–671.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Flack, Robert, and Brian Ross. 2011. "Evolution of Architectural Floor Plans." *Applications of Evolutionary Computation*: 313–322.

Holland, John H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor: University of Michigan Press.

Homayouni, Hoda. 2000. "A Survey of Computational Approaches to Space Layout Planning (1965–2000)." *Department of Architecture and Urban Planning University of Washington*.

Marler, R. Timothy, and Jasbir S. Arora. 2004. "Survey of Multi-Objective Optimization Methods for Engineering." *Structural and Multidisciplinary Optimization* 26 (6): 369–395.

Youmans, Robert J., and Thomaz Arciszewski. 2014. "Design Fixation: Classifications and Modern Methods of Prevention." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28 (2): 129–137.

IMAGE CREDITS

All drawings and images by the authors.

Danil Nagy is a Principal Research Scientist at The Living, an Autodesk Studio. His work and research focuses on computational design, generative geometry, advanced fabrication, machine learning, and data visualization. Danil has spoken and presented work at a variety of conferences and venues, including the Design Modelling Symposium, Biofabricate, Techonomy:Bio, and SIGGRAPH.

Lorenzo Villaggi is an Associate Research Scientist at The Living, an Autodesk Studio. His work focuses on generative design, new materials, and novel forms of visualization. Lorenzo also co-founded and co-edits : (pronounced "colon"), a collective workshop on architectural practices based in New York City. His work has been exhibited in many internationally renowned venues including Milan Design Week and the New Museum.

Dale Zhao is a Research Engineer at The Living, an Autodesk Studio. His work mainly involves the development of in-house applications for numerical optimization, computational design and real-time 3D visualization.

David Benjamin is Founding Principal of The Living, an Autodesk Studio. David has lectured about his work in many parts of the world, and he currently teaches at Columbia Graduate School of Architecture, Planning and Preservation. Before receiving a Master of Architecture from Columbia, he received a Bachelor of Arts from Harvard.