Interactive Architectural Design with Diverse Solution Exploration

Glen Berseth*, Brandon Haworth*, Muhammad Usman*, Davide Schaumann, Mahyar Khayatkhoei, Mubbasir Kapadia, and Petros Faloutsos

Abstract—In architectural design, architects explore a vast amount of design options to maximize various performance criteria, while adhering to specific constraints. In an effort to assist architects in such a complex endeavour, we propose IDOME, an interactive system for computer-aided design optimization. Our approach balances automation and control by efficiently exploring, analyzing, and filtering space layouts to inform architects' decision-making better. At each design iteration, IDOME provides a set of alternative building layouts which satisfy user-defined constraints and optimality criteria concerning a user-defined space parametrization. When the user selects a design generated by IDOME, the system performs a similar optimization process with the same (or different) parameters and objectives. A user may iterate this exploration process as many times as needed. In this work, we focus on optimizing built environments using architectural metrics by improving the degree of visibility, accessibility, and information gaining for navigating a proposed space. This approach, however, can be extended to support other kinds of analysis as well. We demonstrate the capabilities of IDOME through a series of examples, performance analysis, user studies, and a usability test. The results indicate that IDOME successfully optimizes the proposed designs concerning the chosen metrics and offers a satisfactory experience for users with minimal training.

Index Terms—Design exploration, Design optimization, User-in-the-loop

1 Introduction

*

When designing, architects explore a broad set of options to identify the solutions that better satisfy a set of performance criteria while abiding specific constraints [1]. This process is an iterative process whereby design solutions are developed and then progressively tested and refined to maximize the overall design performance [2].

Computer-aided design tools have been developed to help architects address these challenges by predictively analyzing and evaluating the expected performance of a building design [3], [4], [5]. Earlier methods are limited to merely computing quantitative measures for evaluation purposes of a given design option (e.g., in terms of costs, structural stability and energy efficiency). Modern computer-aided design approaches can also produce optimal designs using recent advances in optimization techniques and brute force computing power.

These approaches, however, suffer from two significant limitations. First, with few exceptions [6], [7], they do not account for how people act and interact in these environments. This limitation is arguably one of the most critical design criteria: a building that does not support human needs will likely cause users' dissatisfaction and lack of productivity. Incorporating human movement aspects in an optimization process, however, is very complicated since human factors are hard to quantify.

Second, these approaches tend to produce optimal solutions given a set of encoded constraints, while excluding the designer from evaluating intermediate design options. In the design process, however, both design goals and constraints cannot always be specified beforehand, at the beginning of the optimization process. Due to the ill-structured nature of design problems, design goals and constraints can be discovered in the process of synthesizing new solutions [8]. For this reason, a trade-off should be found between automation and control, whereby designers are actively participating in the optimization process and can contribute to it utilizing *tacit knowledge*–knowledge that is built with practice and can be difficult to communicate or formalize [9].

To address these issues, we propose IDOME, a user-in-the-loop computer-aided design tool that employs architectural optimization with diverse exploration to help architects and designers explore, analyze, and improve their work to maximize human-related parameters. A key aspect of our approach is that the optimization process itself is tuned for exploring alternatives (diversity) rather than merely producing one optimal design at each invocation.

Within IDOME, a user first selects a set of environment elements and specifies associated parameters that may be explored by the system. Then, the user selects one or more metrics to serve as the optimization objectives and defines the regions in the environment where the metrics will be

- *G. Berseth is at the Department of Electrical Engineering and Computer Sciences at UC Berkeley, Berkeley, CA, United States.*
- *B. Haworth, M. Usman, and P. Faloutsos are with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON, Canada.*
- *P. Faloutsos is also with the University Health Network: Toronto Rehabilitation Insititute, Toronto, ON, Canada.*
- *M. Khayatkhoei, D. Schaumann and M. Kapadia are with the Department of Computer Science, Rutgers University, Piscataway, NJ, United States.*

Manuscript received XXXXX; revised XXXXXX.

∗. These authors contributed equally to this work.

computed. IDOME thus generates a set of optimized solutions, one of which will be selected by the user to serve as input for the subsequent optimization iteration.

In this work, we make use of three well-established metrics to capture how people interact with and navigate in an environment: visibility, accessibility, and organization of space [10]. These metrics make up a part of the Space-Syntax suite of tools and methodologies for analyzing environment designs. The system, however, can incorporate other kinds of metrics. A GPU accelerated process addresses the computational costs associated with the repeated calculation of the metrics for large environments.

To solve the optimization process for a diverse set of candidate solutions, we introduce a diversity term in the objective formulation. This diversity term requires the solver to *focus* the search to meet optimality criteria, while simultaneously *broadening* its exploration to maximize the diversity of its candidate solutions. The process of balancing multiple objectives during optimization is a well-known challenge, which is rendered even more difficult by the presence of a diversity term. To address this issue, we propose a hierarchical multi-objective optimization algorithm which balances optimality and diversity while remaining efficient for interactive use without the need for hand-crafted exploration methods.

Our framework can serve in a range of assisting roles, from an efficient way to evaluate alternative configurations which accomplish similar objectives, all the way to a design exploration assistant. We have integrated IDOME within an industry standard architectural design system, Autodesk Revit®. Our results demonstrate the value of our approach to iteratively optimizing and refining architectural design options in a computing-efficient manner. We devise a series of user studies to evaluate the efficacy, usability, preference and usefulness of the proposed approach. To demonstrate efficacy, a user design study showed that subjects using IDOME were able to produce more optimal designs in comparison to subjects who didn't use IDOME, and that users with the diversity exploration choices performed on par with single optima. We evaluate usability with an industry standard usability survey, immediately following a general use design session, which suggests that novices were able to use our system with minimal training and found it usable. We performed two studies to evaluate usefulness of the system. The first was an expert preference survey which showed experts preferred IDOME derived designs. The second was a general use session followed by an expert usefulness survey which suggest that experts found the IDOME approach, the visualizations, and the diversity exploration useful.

Our contributions are summarised as follows:

- We propose a user-in-loop system for computer-assisted exploration of crowd-centric building designs.
- We introduce an efficient hierarchical multi-objective optimization method to balance optimality and diversity of alternative designs without the need for hand-crafted parameter exploration mechanisms.
- We integrate IDOME within the Autodesk Revit® pipeline for demonstration and evaluation.
- We perform in-depth user studies to assess the efficacy, usability, and usefulness of the IDOME approach. These include both large efficacy and usability experiments as well as experiments with experts.

2 Related Work

Computer-aided design (CAD) methods have garnered increasing attention from both researchers and practitioners in recent years, as they allow designers to leverage automation at all stages of the design process. In particular, CAD tools have evolved from being a mere analytic tools to support interactive optimization of building layouts with respect to a wide range of design criteria.

Automated Architectural Design. There is a growing interest in using optimization techniques to explore design spaces for near-optimal solutions given certain problem criteria [11], [12], [13], [14], [15]. Galle [16] focused on exhaustively searching possible layout configurations for small-scale environments. Since then, evolutionary approaches [17], [18] have been used to curb the infeasibility of brute-force methods for larger design spaces. Liu *et al*. [19] introduced functional, design, and fabrication constraints as objective measures to guide the optimization process. Data-driven approaches [20] learn layout configurations from existing databases, which are used to automatically generate new layouts for computer graphics applications. Design objectives can be modelled as forces applied to physical features to generate layout designs automatically [21]. A sophisticated optimization scheme takes into account the visibility, accessibility, and other hierarchical spatial relationships to produce interior design [7], [22] and mid-scale environments [23] configurations. Optimization methods can also successfully account for different physical aspects considered important to architecture such as sunlight [18], materials, energy savings [24] or even acoustics [25].

Interactive Design Solutions. Since early work on methods to assist users to create realistic graphics and animations, interactive methods using evolutionary tools have shown promise [26]. The combination of a graphics interface that enables a more intuitive interaction with the background mathematical structures and evolutionary search methods creates a powerful design framework. While automated approaches can take into account objective criteria, architectural design inherently involves subjective decisions about aesthetics, domain expertise, and hard-to-quantify criteria such as human activity and its relationship to the environment. These challenges are mitigated by proposing computer-assisted, interactive tools that keep the user in the design loop, while using automation to inform the designers decision-making [27], [28], [29], [30], [31], [32], [33], [34]. Harada *et al.* [35] uses shape grammars to support the interactive manipulation of architectural layouts. Recent works have proposed optimisation-based interactive design tools to facilitate furniture arrangement using interior design principles [22], [36]. Akase *et al*. [37] proposed an online room design framework where the objective function entirely relies upon the user's evaluation.

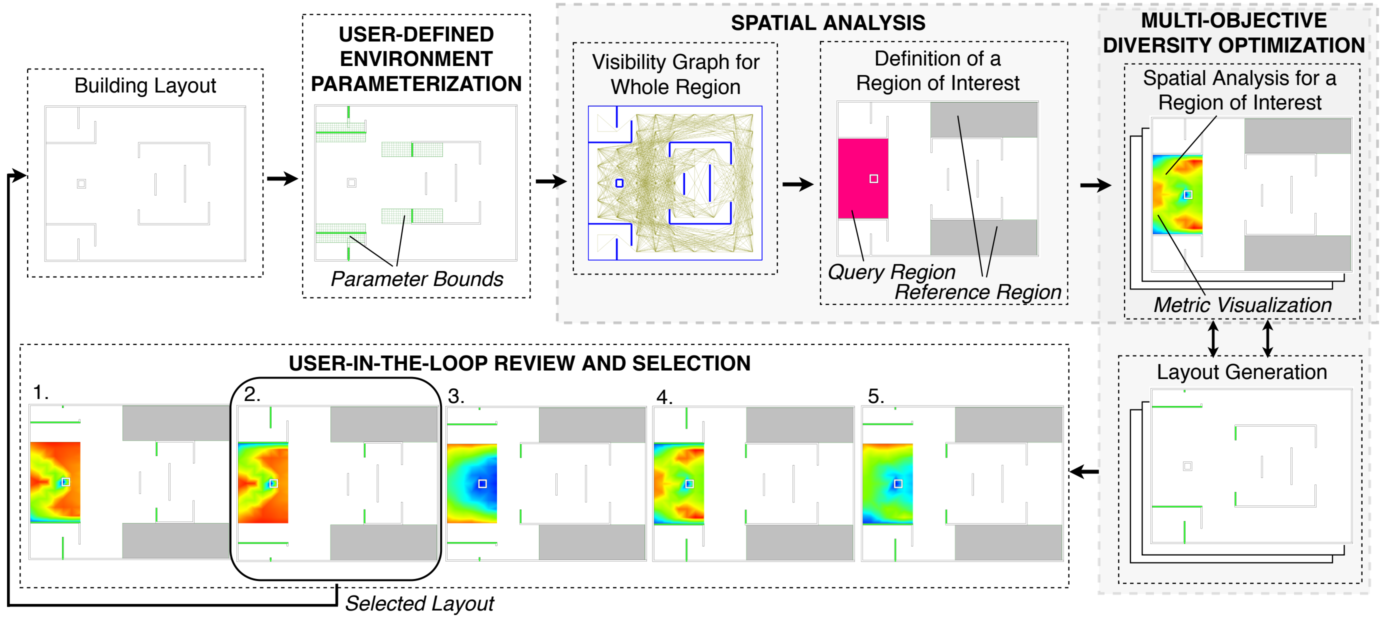


Fig. 1: IDOME Framework Overview. With an initial environment design, the user specifies permissible alterations to the layout as bounds on the degree to which different environment elements can transform. The user then specifies one or more focal regions in the environment for which different spatial measures are computed, to quantify visibility, accessibility, and organization of the space. A multi-objective hierarchical diversity optimization produces a set of diverse near-optimal solutions concerning user-defined optimality criteria, from which the user may select one and repeat the process as desired.

Automatic Exploration of Diverse Designs. To better balance automation and the user's creative control, researchers have proposed approaches for exploring multi-dimensional search spaces to find multiple, diverse, yet optimal solutions which can be provided as suggestions to the designer. This provides the designer with more control, allowing them to harness the power of computation to efficiently explore large design spaces, in domains including multi-body dynamics [38], [39], light selection and image rendering [40]. Introducing diversity for exploration as part of the optimization formulation makes the problem more challenging, with many proposed solutions including constraint programming [41], evolutionary methods [42], and domain-independent methods [43], [44].

Design Metrics. A key challenge in the analysis of environment designs is to account for factors related to its human occupants, which are difficult to quantify. Space-Syntax is an established framework for human navigation related spatial analysis [10], [45], [46], [47]. This approach represents space as different types of graph [48], and then analysis the space by computing a wide range of graph metrics, which have been shown to correlate with human behaviour [49], [50], [51], [52]. While other approaches have been developed for static analysis [53], [54], such approaches require detailed input of user activities, which may not be available in early design stages. Other approaches instead, use dynamic crowd simulations [55]. However, these may require expensive and time-consuming calculations making them ill-suited for incorporation within an optimization algorithm. In this work, we use a set of static measures grounded in the well-established Space-Syntax methodologies [10], [56], [57], [58]. However, our framework is independent of this particular choice and can easily incorporate other spatial measures.

Comparison to prior work. Compared to current approaches, our work strives to keep the user as a central player in the optimization process to inform the design of environments. We also implement a diversity optimization approach that enables fast design optimization while at the same time suggesting to the design a broader set of options that can be selected for future exploration. On top of this, we integrate our approach into a state-of-the-art architectural design software to guarantee an optimal user experience.

3 Overview

An overview of the major components of IDOME is illustrated in Fig. 1. In the following paragraphs we describe each component.

User-Defined Environment Parameterization. Given an initial environment layout, a user first selects elements (e.g., pillars, wall junctions, or walls), and specifies limits on different degrees of freedom of these elements. These attributes represent a user-defined parametrization of the environment layout, which together with the associated limits, determines the set of admissible configurations of the environment. See Section 4 for details.

Spatial Analysis. After constructing a discrete graph representation of free space in the environment, IDOME computes different spatial metrics to quantify visibility, accessibility, and organization of the space. While any metrics may be computed over the environment, these measures are predictive of spatial utilization and human movement, and serve as the basis for our optimization algorithm. Additionally, the user may optionally restrict the computation of these measures to specific regions of interest (e.g. a specific key location in a space, such as a statue in a museum or an emergency exit). See Section 5 for details.

Multi-Objective Diversity Optimization. The environment parameters, designer constraints, and spatial measures are used to formulate an optimization problem over the space of environment configurations. We desire to keep the user central to the design process while using automation to provide multiple suggestions for improving the current design. To facilitate this, our objective formulation generates diverse layouts, while preserving the aforementioned optimality criteria. IDOME efficiently searches through the space of permissible environment configurations to identify diverse, yet near-optimal exploration candidates using a hierarchical multi-objective optimization algorithm. See Section 6 for details.

User-in-the-loop Review and Selection. The designer reviews each of the candidate designs, selecting one to use as the basis for subsequent alterations through a tightly coupled design and optimization process. Using IDOME, designers can leverage computation to account for difficult-to-interpret features such as accessibility and visibility of an environment with respect to its human occupants. The layouts are provided to the user as suggestions, together with visualizations of the spatial measures. The designer may browse these and make an informed decision on which design solution best suits their goals.

4 User-Defined Environment Parameterization

The parametric space representation supports the iterative alteration of design elements for optimization purposes. The architectural elements of a building and their connections can be represented by an undirected architectural graph $G_A = \langle N, E\rangle$, comprising of a set of nodes $N = \{n_i\}$ and edges $E = \{e_i\}$. Each node $n \in R^2$ specifies a location in 2D-space. Each edge is a pair of nodes $e = \langle n_i, n_j\rangle$. An example of a building layout and the associated graph abstraction is shown in Fig. 2. In this representation, walls are edges (e) in the graph, while nodes (n) represent end points and junctions between walls. If a connected component in an architectural graph contains a single node and no edges, such as n_9 in Fig. 2, then the node itself represents an element with fixed structure. The geometry of each element (wall, column, etc.) is stored in a database and associated with the corresponding node or edge.

Given an architectural graph G_A, the user can define the design space by parameterizing and constraining the attributes of selected nodes or groups of nodes. For demonstration, we focus primarily on rigid body transformations of position and orientation.

Each element of the parametrization, $q_i = \langle N_i, p_i, g_i, c_i\rangle$, contains a set of nodes $N_i = \{n_j\}$, a transformation g_i that will be applied to the nodes, the magnitude p_i of the transformation, and the limits or constraints c_i on the magnitude p_i. Grouping the free parameters in a vector $\mathbf{p} = \{p_1, \ldots, p_k\}$ the parametrization of the design space can be compactly represented as $G_A(\mathbf{p})$.

Fig. 2 shows an example of a floor plan with sixteen nodes. The arrows, arc, and painted regions around node n_9 show the user specified range that the node can translate and rotate within. The group of nodes $\{n_1, n_2, n_3\}$ (in red) can rotate around n_2 within the specified range. The nodes n_{12} and n_{13} can move in the $y-$axis but are constrained to maintain their initial distance, forming another group.

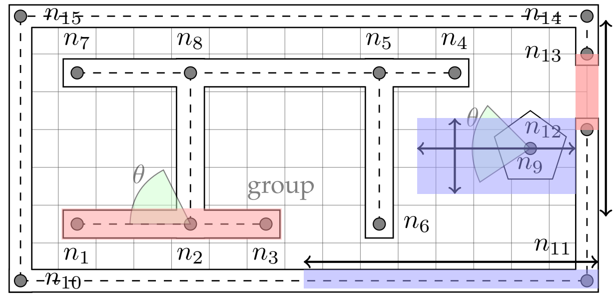
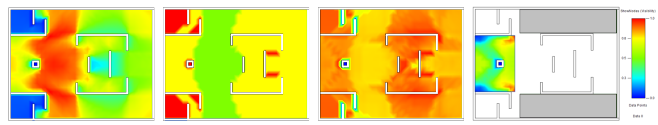


Fig. 2: A floor plan and the corresponding graph parametrization of the walls, doors and other rigid elements. User-selected nodes of the graph, n_i, can be grouped, translated, scaled, and rotated within user defined bounds shown in colour and with arcs and arrows.



(a) Visibility (b) Tree depth (c) Entropy (d) Visibility for *Region of Query*

Fig. 3: Metrics values represented as heatmaps (from blue to red) for a room in the Metropolitan Museum of Art. (a) Degree of visibility, where redder areas have the highest connectivity. (b) Tree depth, where bluer (and greener) areas have higher accessibility (c) Entropy, where redder areas have high entropy (order), resulting in a better spatial organization. (d) Degree of visibility in *Region of Query* with respect to *Region of Reference* which is shown in grey. Notice how changing the reference from the entire environment (as displayed in figure (a)) to just the top and bottom hallways (as displayed in this figure) has affected the values of the metric.

5 Spatial Analysis

The proposed spatial analysis aims to quantify static geometrical attributes of an environment that directly affect how people use the environment. There are several approaches to this [48], [59], however we chose the visibility graph analyses (VGA) from Space-Syntax, as they tend to be more informative than alternative representations, such as axial maps [47], [59]. VGA has been shown to correlate with human movement [60]; to be useful measuring wayfinding tasks [61]; and to be related to wayfinding, movement, and space use [62]. Additionally, recent work has shown that these metrics are good for creating building designs that are human movement aware [63]. However, our method is not restricted to any specific metrics.

5.1 Visibility Graph

To construct a visibility graph, $G_V(V, E)$, we first sample the environment with a finite grid V, and then create an edge, $e \in E$, between every pair of nodes that share an unobstructed line of sight.

Typically, in Space-Syntax Visibility Graph Analysis, every vertex of grid V is a vertex of the visibility graph. In many cases it may be useful to define the visibility graph, and consequently the associated measures, on specific regions of interest. For instance, we may be more interested in the accessibility of certain doors, or the visibility of an exit sign, from specific hallways in the environment. To support this important feature, we allow the user to define two sets of grid vertices, the *Region of Query* with vertices $V_q \subset V$, and the *Region of Reference* with vertices $V_r \subset V$, see Fig. 1. We then construct a visibility graph from these two sets of vertices by computing the lines of sight between the vertices in the *Region of Query* and the vertices in the *Region of Reference*. The user defined regions provide greater flexibility to the user, giving them more control over the spatial queries to be performed on the layout. Putting everything together, the visibility graph depends on the architectural graph, its parametrization, and the regions of interest:

$$G_V = \phi(V_r, V_q, G_A(\mathbf{p})). \quad (1)$$

The spatial measures described in the next section are computed only for the vertices of the region of query.

5.2 Metrics

Given a visibility graph $G_V(V, E)$, we compute selected metrics that characterize meaningful relationships between floor plans and human behaviour. Examples are provided in Fig. 3.

Degree of Visibility. The degree of visibility, k_i, of a vertex $v_i \in V$ is the number of edges incident to the vertex, in other words the number of its immediate (1-hop) neighbours N_i. Regions with high degree of visibility can be considered to be more connected, safe, or important [10], [56].

Tree Depth. Let $G_V^i \subseteq G_V$ be the largest connected component that contains vertex $v_i \in V$. The minimum height Trémeaux tree rooted at v_i is the tree depth, d_i. Tree depth has a few intuitive interpretations. First, it measures how far G_V^i is from being a star [64]. Second, a vertex with large tree depth is connected to other regions of the environment through a long sequence of vertices. Thus, tree depth often relates to the notion of *accessibility* in an environment [56]. Tree depth values, together with context dependent information, allow a user to make flow and congestion estimations on specific areas of a layout.

Entropy. Let $G_V^i \subseteq G_V$ be the largest connected component that contains vertex $v_i \in V$. Given a Trémeaux tree T_i rooted at vertex i with n_i^j vertices at each level j, a probability distribution $p(j|i)$ is constructed for T_i over the domain $j \in [1, |V_i|]$, where V_i is the set of vertices in G_V^i, and through this distribution we define the entropy h_i at vertex i as follows:

$$p(j|i) = \frac{n_i^j}{\sum\limits_{j' \in L_i} n_i^{j'}},$$
$$h_i = -\sum_{j \in L_i} p(j|i) \log_2 p(j|i). \quad (2)$$

Technically, $p(j|i)$ is the probability that a vertex in V_i will be at level j of the tree T_i. In more intuitive terms, entropy measures the *organization* of an environment. Low entropy at a vertex means that the decision tree rooted at the vertex is unbalanced, or in other words the branching factor varies widely from level to level. This unbalance can materialize both as bottlenecks or areas with too many options which may disorient a person moving through the associated areas. In some sense, while tree depth relates to path lengths, entropy relates to the uniformity of the paths: the higher the entropy, the more uniform the branching, and thus better organization. Typically higher uniformity affords easier pedestrian decision making and navigation [56], [65].

These metrics thus far are computed from a single vertex in the G_V. To get a metric value over the entirety of the design we average the metric over every vertex in the G_V.

$$K(G_V) = \bar{k_i}, \quad D(G_V) = \bar{d_i}, \quad H(G_V) = \bar{h_i}. \quad (3)$$

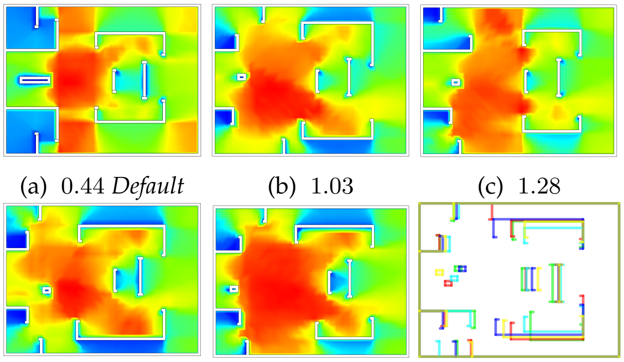
5.3 Metrics Parallelization

The aforementioned metrics can be computationally expensive. In order to mitigate this computational overhead, we off-load the construction of the visibility graph and the forest to the GPU. Transforming tree construction from a sequential process to a parallel one, is the main challenge in this context. We can accelerate the forest construction rather than just the single tree construction. The GPU implementation is orders of magnitude faster than the serial one. Performance numbers are shown in Section 7.2. See the supplementary document for more details on the algorithm.

6 Multi-Objective Diversity Optimization

Hierarchical multi-objective optimization methods optimize one objective at a time, in order, in a fashion similar to coordinate descent. Each optimized objective becomes part of the objective function in the form of a soft constraint for the optimization of the next objective. A hierarchical approach appears to be the most practical approach for this problem space. It allows for more practical and intuitive control of the trade-off between optimality and diversity, in the form of a lower bound with respect to the optimal solutions.

Similar optimization problems have been solved in the graphics literature with a combination of Simulated Annealing, Metropolis-Hastings and exploration heuristics [22], [36], with parameters adjusted both manually and automatically for typical furniture layouts. The convergence rates of these methods, however, can make them prohibitive for interactive systems. Literature in engineering and optimization seems to conclude that Covariance Matrix Adaptation (CMA) [66] converges faster than competing methods in many cases, and that it is probably the most appropriate method for automated design problems [67]. Our approach is built on evolutionary techniques for producing optimized designs, specifically the CMA optimization technique. However, the evolutionary concepts are abstracted away from the end user, where interaction in our system is facilitated through selection from diverse exploration candidates. For more details see the supplementary document. Thus, in our work, we formulate and solve a hierarchical optimization



(a) 0.44 *Default* (b) 1.03 (c) 1.28

(d) 1.22 (e) 1.46 (f) *Comparison*

Fig. 4: Example collection of diverse candidates. (a) is the default layout, (b-e) are the five members of the diversity set, and (f) All solutions to better illustrate their differences.

problem based on CMA without the need for additional exploration heuristics.

The rest of this section presents in detail our optimization formulation, starting with the objective that includes both user supplied constraints, and the measures that we discussed earlier.

6.1 Design Constraints

In the context of architectural optimization, a user may wish to impose a number of conditions on specific design elements, such as a minimum amount of open space in passages, aesthetic relationships, or building codes. In the following paragraphs we detail the constraints considered in this study.

Clearance. A measure of open space between architectural elements, clearance is computed as the aggregate Minkowski sum of each wall and a disk D_r of radius $0.5\ m$, which approximates the minimum width of a hallway. The Minkowski sum between a polygon and a disk dilates the polygon, effectively adding a buffer area around an obstacle or wall for comfortable passage.

$$clr(G_A) = \sum_{e_i, e_j \in G_A} A((e_i \oplus D_r) \bigcap (e_j \oplus D_r)), \quad (4)$$

where $A(\cdot)$ computes the area, and $\oplus$ denotes the Minkowski sum between two polygons and $\bigcap$ is the geometric intersection of the two Minkowski sums. Adjoining walls are excluded from this computation. The associated penalty function is $g_{clr} = clr(G_A)^2$.

Total wall length. The sum of wall lengths in the *new* environment (G_A^n) with respect to the original environment (G_A^o) is $wall(G_A^n, G_A^o) = |\mathcal{S}(G_A^n) - \mathcal{S}(G_A^o)|$, where $\mathcal{S}(\cdot)$ computes a sum over the length of every edge (or wall) in the graph. This penalty function is used to constrain the repositioning of elements to not reduce or increase the quantity of wall surface area in an environment. This particular penalty method is appropriate for museums and art galleries where there is a desired amount of wall surface area needed to display an art collection.

6.2 Diversity Objective

Unlike a standard optimization approach that produces a single design solution $\mathbf{p}^*$, IDOME produces a set of optimal solutions $\mathcal{D}^* = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$ whose members differ from each other. For efficiency, instead of augmenting the parameter vector $\mathbf{p}$ with additional elements for each member of the diversity set, a round robin technique is used, where one member in $\mathcal{D}$ is optimized at a time while keeping the other parameters members constant.

In practise, enforcing diversity can still lead to a clustering of solutions [39]. To avoid clustering, we impose a minimum distance between members of $\mathcal{D}$ which is defined as follows:

$$div(\mathbf{p}_m, \mathcal{D}) = k(\sum_{j \in \mathcal{D}} dn(\mathbf{p}_j, \mathbf{p}_m)) - k_m\, d(\mathbf{p}_m, \mathcal{D}), \quad (5)$$

$$d(\mathbf{p}_m, \mathcal{D}) = (\min(0, \min_{j \in \mathcal{D}, j \neq m}(dn(\mathbf{p}_j, \mathbf{p}_m)) - d_{min}))^2, \quad (6)$$

where $dn(\cdot,\cdot)$ normalizes its arguments over the parameter constraints before computing their Euclidean distance, and $d(\mathbf{p}_m, \mathcal{D})$ is the minimum distance between $\mathbf{p}_m$ and all other members in $\mathcal{D}$. Equation 6 ensures that diverse members don't cluster by adding a cost when the closest neighbours are less than d_{min} away. The terms d_{min}, k and k_m are experimentally determined hyper-parameters that control the influence of the diversity term.

6.3 Hierarchical Optimization Formulation

For a set of optimal solutions, $\mathcal{D}$, the objective vector is aggregated over the entire set. This results in the following multi-objective optimization problem:

$$\mathcal{D}^* = \underset{\mathcal{D} \subset \mathcal{P}}{\arg\max} \sum_{\mathbf{p} \in \mathcal{D}} \langle -\, g(\mathbf{p}), K(\mathbf{p}), -D(\mathbf{p}), H(\mathbf{p}), div(\mathbf{p}, \mathcal{D}) \rangle, \quad (7)$$

$$\text{s.t.} \quad C(\mathbf{p})$$

where $C(\mathbf{p})$ are the parameter bounds specified by the user and $g(\mathbf{p})$ is a combination of constraints described in Section 6.1. Solving this problem ideally produces a set of solutions with maximum spatial objectives, minimum penalties, and maximum diversity.

We solve the above problem as follows. For each objective we specify an order (ranking) and a desired minimum improvement threshold z. The desired threshold z is a number between [0, 1], and specifies the minimum percentage of optimality that the process must maintain for the specific objective while optimizing other objectives. For example if an objective is ranked first with a threshold of 0.7, then the optimization process will optimize it first, ignoring other metrics. After converging to an optimal value, a constraint is added to the second objective that imposes a penalty if the previous objective falls below 70% of its optimal value. The process repeats for all objectives with the near-optimality constraints given.

Algorithm 1 describes the hierarchical multi-objective optimization method over the objective vector

$$\mathbf{f} = \langle -\, g(\mathbf{p}_0), K(\mathbf{p}), -D(\mathbf{p}), H(\mathbf{p}), div(\mathbf{p}_0, \mathcal{D}) \rangle. \quad (8)$$

Algorithm 1 Hierarchical Multi-Objective Optimization

1: **Input:** Number of diversity members, n
2: **Input:** Vector of objective thresholds, $\mathbf{z} = \langle z_i \rangle$
3: **Input:** Initial parameter vector, $\mathbf{p}_0$
4: **Input:** Vector of objective functions, $\mathbf{f} = \langle f_i \rangle$
5: **Input:** Variance, σ, Sample size, λ
6: $\mathcal{T} \leftarrow \{\}$
7: **for each** $f_i \in \mathbf{f}$ **do**
8: $\quad \mathbf{\Sigma} \leftarrow \sigma^2 I, \mathbf{p}_{opt} \leftarrow \mathbf{p}_0$
9: $\quad$ **while** $\neg$ *Terminate*() **do**
10: $\quad\quad$ **for each** $j \in (1, \lambda)$ **do**
11: $\quad\quad\quad \mathbf{p}_j \leftarrow \mathcal{N}(\mathbf{p}_{opt}, \mathbf{\Sigma})$
12: $\quad\quad\quad y_j \leftarrow f_i(\mathbf{p}_j) - g_h(\mathbf{p}_j, \mathcal{T})$
13: $\quad\quad \langle \mathbf{p}_{opt}, \mathbf{\Sigma} \rangle \leftarrow Update(\mathbf{p}_{opt}, \mathbf{\Sigma}, \langle y_j \rangle, \langle \mathbf{p}_j \rangle)$
14: $\quad l \leftarrow f_i(\mathbf{p}_0) + z_i \cdot (f_i(\mathbf{p}_{opt}) - f_i(\mathbf{p}_0))$
15: $\quad \mathcal{T} \leftarrow \mathcal{T} \cup t(l, \infty, \cdot)$
16: $\mathcal{D} \leftarrow DivOpt(n, g_h(\cdot, \mathcal{T}), \mathbf{p}_0)$
17: **return** $\mathcal{D}$

It is important to keep the diversity metric as the last objective. The diversity metric considers a set of diverse members, while the other metrics operate over a single member of the set. It can be necessary to constrain the optimization using penalty functions. To facilitate these constraints we put the penalty function(s) first in the hierarchy.

To incorporate an objective as a constraint during the hierarchical optimization process, a *threshold function* is used. These functions are constant or simply zero when the input is within a given range, and rapidly increase when the input is outside the range:

$$t(l, u, x) = \begin{cases} (l - x)^2 & \text{if } x < l \\ (x - u)^2 & \text{if } x > u \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

For a set of threshold functions $\mathcal{T}$ the total threshold violation cost is

$$g_h(\mathbf{p}, \mathcal{T}) = \sum_{t \in \mathcal{T}} t(\mathbf{p}). \tag{10}$$

For each objective, a CMA-based optimization is performed (lines 9-15). At the end of an individual objective optimization, a threshold constraint is created and added to the vector of threshold constraints $\mathcal{T}$ (lines 16-17). At the end of the main loop, the optimal parameter vectors are captured within the thresholding function vector, $\mathcal{T}$. The last objective, diversity, is optimized using *DivOpt*() in Algorithm 2, which searches for a diverse set of near optimal solutions given the set of threshold functions constructed. Note that the other objectives are now represented as penalties through the threshold functions. The next section describes the final step of our hierarchical optimization, the diversity objective.

6.4 Diversity Optimization

A round-robin method is used to select and optimize each diversity member m one at a time, see Algorithm 2. Each member is initialized using $\mathbf{p}_0$ and progressively diverge from each other as the optimization unfolds. In each round,

Algorithm 2 Diversity Optimization:

1: **function** *DivOpt*$(n, f, \mathbf{p}_0)$
2: **Input**: Number of diversity members, n
3: **Input**: Objective function, f
4: **Input:** Initial parameters, $\mathbf{p}_0$
5: **Given:** Variance, σ, Sample size λ
6: **for** $i \in (1 \dots n)$ **do**
7: $\quad \mathbf{\Sigma}_i \leftarrow \sigma^2 I, \mathcal{D}_i \leftarrow \mathbf{p}_0$
8: **while** $\neg$ *Terminate*() **do**
9: $\quad$ Choose m from $P(1 \le x \le n)$
10: $\quad$ **for each** $i \in (1, \lambda)$ **do**
11: $\quad\quad \mathbf{p}_i \leftarrow \mathcal{N}(\mathcal{D}_m, \mathbf{\Sigma}_m)$
12: $\quad\quad y_i \leftarrow div(\mathbf{p}_i, \mathcal{D}) - f(\mathbf{p}_i)$
13: $\quad \langle \mathcal{D}_m, \mathbf{\Sigma}_m \rangle \leftarrow Update(\mathcal{D}_m, \mathbf{\Sigma}_m, \langle y_i \rangle, \langle \mathbf{p}_i \rangle)$
14: **return** $\mathcal{D}$

a single member is selected from $\mathcal{D}$ and candidate parameters are sampled using CMA [66]. A simple in-order method is used to select the next member in each round. More complex, or random, selections may be employed, but we empirically found this strategy to work well. In lines 10-11, the objective values for those candidates are calculated. In line 12, the structures in CMA that influence the optimization evolution are updated. The termination condition is dependent on the optimization progress with respect to the improvements made on $\mathbf{p}_{opt}$ and $\hat{y}^*$ and the maximum number of function evaluations, which are parameters of CMA [66].

7 Applications and Performance

In this section we demonstrate different applications of IDOME and we evaluate its performance. First, we provide three examples of interactive optimization (Section 7.1). Then, we evaluate its computational performance (Section 7.2).

7.1 Example of interactive optimization

We demonstrate the application of IDOME in three selected scenarios.

Maze. In this example, we use IDOME to alter the complexity of a maze-like environment. We demonstrate how our method is able to remove or introduce complexity in an environment by altering the objective definition, while producing several diverse designs that meet user-defined parameterization criteria. The results are illustrated in Fig. 5. First, we generate a series of design options that *minimize* the environment's complexity: starting with a standard maze, we maximize the visibility, minimize the tree depth (which maximizes accessibility), and minimize the entropy (which maximizes order). The diverse set of exploration layouts align the doorways to minimize long-windy passageways which have higher tree depth (b,c). Then, we modify the objective function to *maximize* the environment's complexity. To do so, we feed the more ordered environments (b) into the system with the objective measure inverted to minimize $-F(\cdot)$. The resulting diverse layouts (d,e,f) are of similarly complexity to the original maze, while providing variations of the original design.

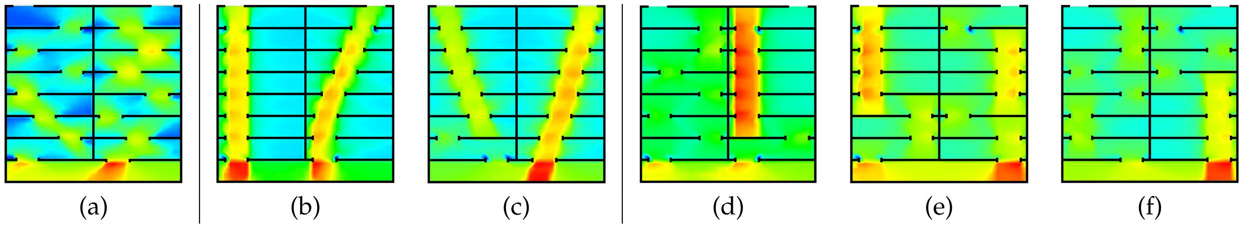


Fig. 5: Maze design. diversity set of maze-like designs of size 20x20m approximately with combined spatial metrics displayed in the form of heat maps. Figure (a) displays an initial maze configuration. Figures (b) and (c) show optimized maze to reduce environment complexity. The result from (b) is optimized to increase complexity and produce new mazes illustrated in figures (d), (e) and (f).

Art Gallery. In this example, we demonstrate the diversity optimization ability to generate variable design options for a gallery space of the Washington Art Museum (Fig. 6). Here, the space was parametrized to allow for alternative configurations of the gallery corridor and exposition rooms. IDOME produced various alternatives to an original design to maximize visibility, connectivity and spatial organization.

Metropolitan Museum of Art. In this example, we tested IDOME in a significantly larger environment compared ones previously considered. Namely, we analyzed the Metropolitan Art Museum with respect to visibility. Based on the analysis' results, we identified a specific area of the building (in the top-right corner) with reduced visibility. We used IDOME to find options with improved visibility while maintaining the amount of wall surface area, which is necessary for displaying works of art. Fig. 7 visualizes the optimized layout with respect to the whole building (in the top part) and only the considered area (in the bottom part). In the figure, we highlighted key areas of improvement, which involved (a) enlarging the passage between Zone 1 and Zone 2 to create a centralized exposition space, which is better connected to the adjacent zones and to the previously disconnected Zone 3; (b) improving the connection between Zone 3 and 4 by enlarging the width of the zones and the passage between them; (c) transforming Zone 5 from a narrow corridor to a larger exposition space better connected with the adjacent zones; and (d) improving the visual connectivity between Zone 7 and 8 among themselves and with respect to Zone 9 and 10.

7.2 Performance Analysis

Spatial Metrics. Fig. 8 illustrates the comparative performance of our spatial analysis framework (Section 5) using single-threaded CPU, multi-threaded CPU, and GPU implementations. It is evident that the GPU implementation completes the computation much faster. For example, on a grid of 900 vertices, over an effective area of $3600\ m^2$ using a 0.5 cell per meter visibility graph granularity, the GPU takes $10\ ms$ compared to the CPU's $300\ ms$ (4-threads) on average to generate the visibility graph, construct the corresponding forest, and calculate the objective. This advantage increases as the number of vertices in the graph increases, with an order-of magnitude speedup. This test compares Intel Xenon at 3.5 GHz with Nvidia GeForce GTX 1070.

Diversity Optimization. Table 1 provides the computation times of diversity optimization for four exemplar environments. The results indicate that it takes few seconds for IDOME to produce diverse solution candidates in moderately complex designs (hundreds of vertices in the visibility graph). Several seconds are required to produce diverse candidates in significantly more complex environments, as the one illustrated in Fig. 7. While IDOME supports the optimizations of environments at different scales, a trade off exists between environment complexity and optimization speed.

Optimization Convergence. The convergence or stopping conditions of the optimization algorithm has a dramatic impact on the computational performance as well as the quality of the results. The default termination conditions are overly conservative for this application, leading to long optimization times with negligible effects on quality after the first few iterations [66]. The termination conditions are adjusted to return results after the optimization has converged to $\sim\ 95\%$ of optimal. This leads to significant performance gains.

8 User Studies

In this section, we present four user studies to assess the efficacy, usability, preference, and usefulness of IDOME during design tasks. The efficacy study examines how well novice users perform with respect to baseline optimization approaches. The usability study solicits novice perception and opinion of the usability of the system and approach. The preference study solicits experts' preferences with respect to IDOME derived designs. The usefulness study solicits their perception and opinion of the usefulness of the system and approach.

General Materials and Methods. All participants used a Desktop PC (Windows 7 64-bit, 8GB RAM, AMD FX(tm)-8320, 8 Computer Cores, 3.5GHz). Using a simple room as a teaching tool, the participants were given short instructions on how to manipulate and set parameter optimization bounds for translation and rotations of environment elements. Participants were then shown how to explore and select candidates from the diversity set.

The spatial metrics considered in these studies were explained to the participants in general terms (plain language), i.e. Degree of Visibility, Tree Depth, and Entropy are translated to visibility, accessibility, and organization respectively. Participants were told that: visibility related to how visible any portion of the environment may be to another; accessibility related to how accessible the environment is; and organization related to how confusing the layout of the

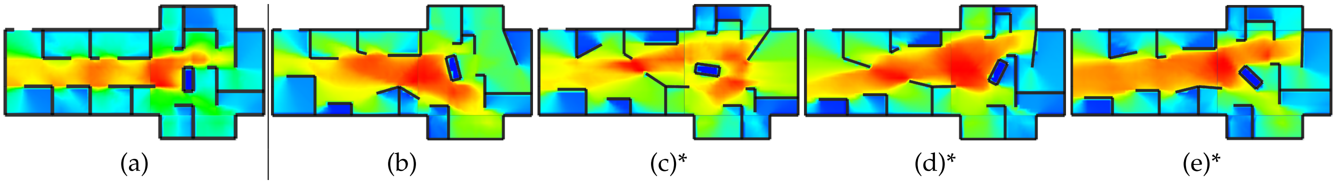


Fig. 6: Art gallery design. The diverse set of art gallery designs with combined spatial metric values displayed as heatmaps over the entirety of each design. Figure (a) is the original art gallery design. Figures (b-e) show the diversity members provided by the IDOME system for a particular environment parametrization. The (*)s identify the designs that six expert architects independently designated as preferred.

Environments	Ref Vertices	Query Vertices	Total Vertices	Effective Size (m^2)	Objective Calls (c)	Graph (ms/c)		Forest (ms/c)		Penalty (ms/c)	Total Time (s)	
						CPU	GPU	CPU	GPU	CPU	CPU	GPU
Simple Room (Fig. 10 right)	361	25	361	1444	692	3.9	0.88	0.7	0.56	0.08	4.35	**2.25**
Large Room (Fig. 10)	1369	81	1369	5476	692	61.08	1.68	19.29	1.76	0.12	57.26	**3.62**
Exposition Room (Fig. 4)	588	208	588	2352	772	23.37	1.04	15.98	1.91	1.06	34.04	**5.06**
Art Gallery (Fig. 6)	487	438	915	3660	732	67.59	1.35	22.73	1.93	5.36	72.69	**7.41**

TABLE 1: Diversity optimization running times. These results were computed using GeForce GTX 1070 and Intel Xenon at 3.5 GHz on a range of environments from simple and small to large and complex. Note that while the system is not real-time, it is sufficiently fast for interactive use. Sampling ratio is 0.5 cell/meter.

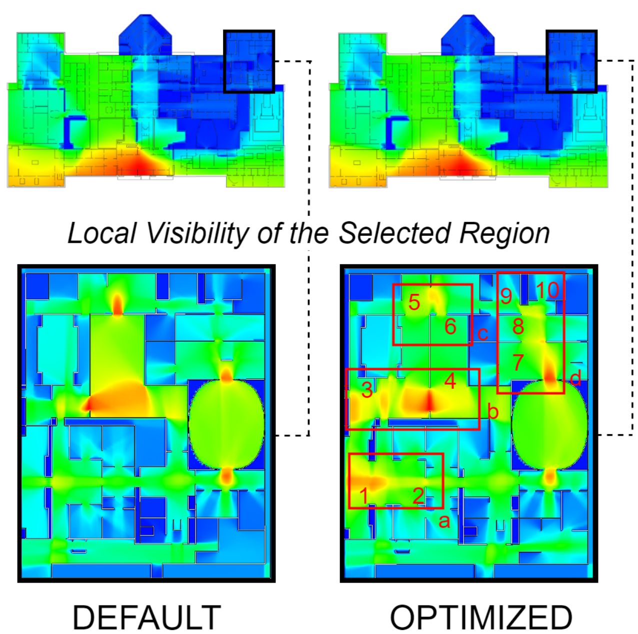


Fig. 7: Metropolitan Museum Design. We illustrate a visibility analysis of the whole museum and the optimization of a specific are of the layout with reduced visibility. We highlight the major improvements produced by IDOME: (a) enlarging the passage between zone 1 and zone 2 to create a centralized exposition space, which is better connected to the adjacent zones; (b) enlarging the size of zone 3 and 4 and improving their connection; (c) transforming zone 5 from a narrow corridor to a larger exposition space better connected to the surrounding zones; (d) improving the visual connectivity between zone 7, 8, 9 and 10.

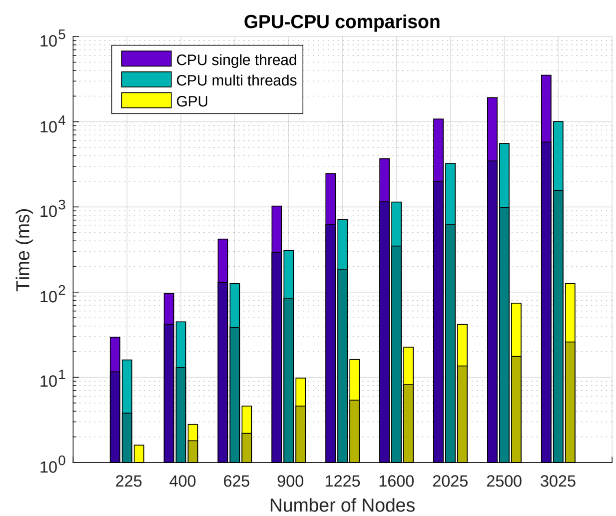


Fig. 8: Spatial analysis framework performance analysis of CPU and GPU implementations. Bars show the total time to calculate the three objectives. Darker and lighter shades depict the time for graph generation and forest construction respectively. Time is in base 10 logarithmic scale.

environment would be for navigation or spatial cognition tasks [56].

8.1 Comparison to Baseline Design

This experiment examines the efficacy of the IDOME method from a design perspective. The hypothesis is that IDOME based designs (participants using the full diversity optimization) outperform manual designs (participants using an industry standard tool) as well as automatically optimized designs (participants using a black box optimization tool that returns a single optimal candidate) for the given tasks and metrics.

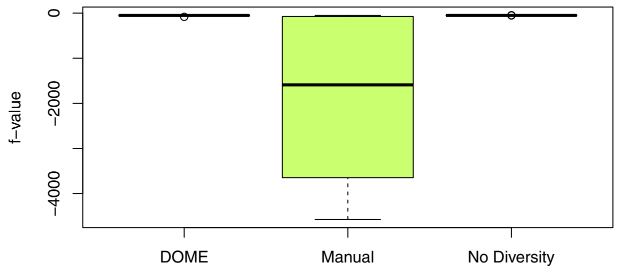
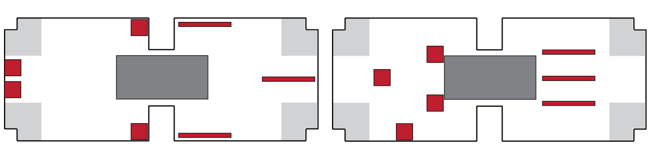


Fig. 9: Comparison of objective values between design tools for the baseline study in Section 8.1. In this task, IDOME helped produce designs with consistently higher objective measures than manual design, and on par with single outcome optimization.

For this study, 18 people volunteered and gave informed consent who were between the ages of 23 to 30 and self identified as 11 males and 7 females. All participants were graduate level students in computer science or a closely related field with little or no environment/architectural design experience.

Materials and Methods. Participants were presented with a complex real world Art Gallery environment. The environment's regions of interests were set up as in a two part art gallery exhibit with an asymmetric set of parametrizable objects (four square pillars, and three walls). The environment itself was a symmetric set of exhibits connected by a small hallway. Participants were given up to 10 minutes to make as many adjustments as they wished to the environment. Each participant was allowed to finish at any point within the 10 minutes, concluding their design when satisfied. Specifically, participants were tasked with modifying the environment to improve accessibility, visibility, and organization of the exhibits in the art gallery while leaving wall clearance potential displays. The experiment was a between subjects design with the three design approaches as primary factors.

Results. Boxplots for the objective values for all tools are shown in Fig. 9. A Kruskal-Wallis test showed a significant difference present amongst the groups ($chi - squared = 8.1871$, $df = 2$, $p = 0.01668$). Post-hoc Conover's and Dunn's tests revealed a significant difference ($p < 0.05$) between IDOME and manual as well as automatic optimization and manual design methods. There was no significant difference between the automatic optimization and IDOME methods.



Manual (a) IDOME-assisted (b)

Fig. 10: Selected layouts from the user study. Light grey regions form the *Region of Interest*. Dark grey regions form the *Region of Query*. In red, the design elements considered in this study. Note the different final design configurations produced by (a) manual and (b) IDOME assisted designs.

Discussion. This task requires the participants to solve a combinatorial placement problem of asymmetric objects in a symmetric environment. IDOME produces great results with much higher objective values than manual design.

It is also noteworthy that the variance is much less when using IDOME. These results suggest that manual optimization can be very inconsistent among different participants, while our system can effectively guide the participant and keep the design exploration more focused. Furthermore, this could be a sign that using diversity helps avoid local minima in the design space. It is equally important to note that solutions returned by the IDOME group were diverse in their approaches, with different participants finding new ways to minimize the objective. In contrast to the automatic optimization method, the diversity set allows even novice users to preserve their unique preferences while producing designs that are high value.

8.2 System Usability

In this study, participants from Section 8.1 were asked to complete a usability experiment in which they make use of IDOME. The participants were tasked with increasing the visibility, accessibility, and organization of the environment for a fixed amount of time (15 minutes). Immediately after the design task, participants were given a System Usability Scale (SUS). The SUS is a well-established and tested method for evaluating the usability of a product or system [68].

Results. The summary statistics of the SUS scores are reported in Table 2. The quartiles for the SUS scores are reported in Table 3.

Count	Mean	Median	Standard Deviation
18	70.83	73.75	14.70

TABLE 2: Summary statistics for SUS results, where the score range is from 0 to 100.

Quartile	Range
$< Q_1$	22.5 - 68.12
$[Q_1, Q_2]$	68.12 - 73.75
$[Q_2, Q_3]$	73.75 - 76.87
$> Q_3$	76.87 - 87.5
IQR	8.75

TABLE 3: SUS quartile ranges. The ranges for each quartile of the data are reported to show distribution of the results. The Interquartile Range (IQR) is also reported.

Discussion: The SUS score is a composite measure of usability for a system which has been tested on a variety of tasks and proved to be robust and reliable [69]. SUS scores are scaled to the range of 0 and 100, with scores higher than 68 considered above average and acceptable [69].

The results show that the 18 participants mean and median scores fall within the adjective range of "good" and "excellent" [70]. Furthermore, quartile ranges $> Q_2$ show a strong preference for a high SUS score. This can be interpreted as meaning the IDOME system is highly usable with a degree of confidence.

8.3 Expert Evaluation of Optimization

In this study, we solicit preferences of architects with respect to IDOME derived designs. For this experiment, a roundtable of six professional architects was provided a single preference selection survey for the design variants in Figure 6. This survey provided the blueprint versions of the layouts in Figure 6 for selection, without the heatmap data or reference to their design origin.

Results: The expert participants all chose IDOME derived diversity variants overt the default design. Interestingly, the preferences were evenly distributed over three diversity set members with two votes each. That is, the results show that expert prefer diverse candidates which align with their subjective preferences. Together this shows the importance of the IDOME approach in production environments where architects balance subjective and objective needs and find value in diverse design directions. Additional, rendering is provided in the Supplementary Material.

8.4 Expert Evaluation of Usefulness

In this study, we solicit expert opinion's/perception's of the usefulness of the IDOME approach. Each expert was asked to improve the metrics, given their plain language descriptions, with respect to the environment in Figure 6a using the IDOME tool and was given up to 20 minutes to make as many changes as they wished. Immediately after the session, experts were provided with an in-depth survey and opportunity to discuss the usefulness and their opinions of the IDOME approaches. These questions included both likert-scale responses, simple binary responses, and short answer responses.

For this study, three expert architects from three different architectural firms participated–their demographics information can be found in the Supplementary Materials.

Results. The results provide insight into the usefulness of the IDOME method from the expert perspective. In summary, a diverse set of practising experts found the approach useful, helpful, and valuable. The experts recognized that the system is a research prototype and gave strong preference for this type of participatory intervention. Results for the Liket-scale questions can be seen in Figure 11.

The participants felt strongly that: heatmaps were a preferable method of visualization for these metrics and more (3/3); that modelling human occupancy and behaviour is important (3/3); that they would like a system to automatically identify issues in their design (3/3); and that they like a system that automatically suggests improvements to their design (3/3). All participants, agreed that suggestion and improvements in the design should be provided in a non-prescriptive and unobtrusive way as in the IDOME system (providing diversity sets as possibilities). For example, one participant noted a preference, "with options, as done in the example ." Additional details can be found in the Supplementary Materials.

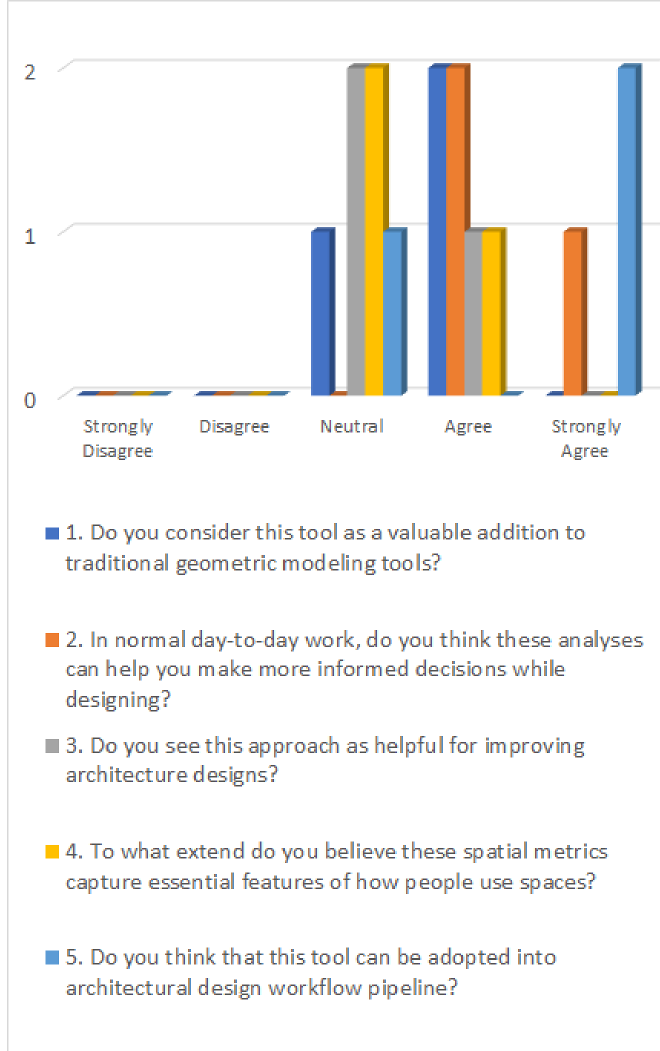


Fig. 11: Results for the likert-scale questions of the expert evaluation of usefulness survey.

9 Conclusion

We have presented IDOME, a user-in-the-loop system for computer-aided design that optimizes environments with respect to human navigation-based metrics. Different from previous approaches, IDOME interactively engages the designers, who need to choose which design solution to further explore. Such design exploration solutions are generated with a diversity optimization algorithm that produces near-optimal solutions with respect to efficiently computed spatial metrics and user-defined constraints. The user studies indicate that IDOME help users produce higher value designs with respect to the chosen set of metrics.

Limitations and Future Work. Like most multi-objective optimization frameworks, our approach includes a variety of weights that the user can set to tweak the results. Although one can rely on default values, it might be beneficial for the user to adjust them. We plan to study the effects of these parameters on the resulting configurations with a large scale experiment, and attempt to identify specific relationships which might serve as guidelines. Different space parameterization system could also be explored in the future [71]. Current approaches in architectural design provide well-established frameworks for space parameterization that could be incorporated within IDOME. Explicit representation of evolutionary parameters in an interactive system for computer-aided design is another promising avenue of future exploration. We acknowledge that we have tested the system for moderate scale design. Further work is needed to test its value for environments of larger scale. We have also identified possibilities for improving performance, such as employing approximate and incremental algorithms for computing the spatial analysis metrics, which we plan to investigate in the future.

Acknowledgments

This is research has been partly funded by ORF/ISSUM, NSF IIS-1703883, NSF S&AS-1723869, NSERC Discovery Grant and the Murray Postdoctoral Fellowship.

References

[1] Y. E. Kalay, *Architecture's new media: Principles, theories, and methods of computer-aided design*. MIT Press, 2004.

[2] H. Rittel, "Some principles for the design of an educational system for design," *Journal of Architectural Education*, vol. 26, no. 1-2, pp. 16–27, 1971.

[3] X. Zhang, D. Schaumann, B. Haworth, P. Faloutsos, and M. Kapadia, "Coupling agent motivations and spatial behaviors for authoring multiagent narratives," *Computer Animation and Virtual Worlds*, vol. 30, no. 3-4, p. e1898, 2019. [Online]. Available: https://www.onlinelibrary.wiley.com/doi/abs/10.1002/cav.1898

[4] D. Schaumann, S. Moon, M. Usman, R. Goldstein, S. Breslav, A. Khan, P. Faloutsos, and M. Kapadia, "Toward a Multi-Level and Multi-Paradigm Platform for Building Occupant Simulation," in *Proceedings of the Symposium on Simulation for Architecture and Urban Design*. Atlanta: Society for Computer Simulation International, 2019, pp. 169–176.

[5] D. Schaumann, S. Breslav, R. Goldstein, A. Khan, and Y. E. Kalay, "Simulating use scenarios in hospitals using multi-agent narratives," *Journal of Building Performance Simulation*, vol. 10, no. 5-6, pp. 636–652, Nov. 2017. [Online]. Available: http://dx.doi.org/10.1080/19401493.2017.1332687

[6] B. Haworth, M. Usman, G. Berseth, M. Khayatkhoei, M. Kapadia, and P. Faloutsos, "Code: Crowd-optimized design of environments," *Computer Animation and Virtual Worlds*, vol. 28, no. 6, p. e1749, 2017.

[7] D. Nagy, D. Lau, J. Locke, J. Stoddart, L. Villaggi, R. Wang, D. Zhao, and D. Benjamin, "Project discover: An application of generative design for architectural space planning," in *SimAUD 2017 Conference proceedings: Symposium on Simulation for Architecture and Urban Design*, 2017, pp. 1–7.

[8] H. W. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy sciences*, vol. 4, no. 2, pp. 155–169, 1973.

[9] D. A. Schön, *Educating the reflective practitioner*. Jossey-Bass San Francisco, 1987.

[10] S. Bafna, "Space Syntax: A Brief Introduction to Its Logic and Analytical Techniques," *Environment and Behavior*, vol. 35, no. 1, pp. 17–29, 2003. [Online]. Available: http://eab.sagepub.com/cgi/content/abstract/35/1/17

[11] P. Block, J. Knippers, N. J. Mitra, and W. Wang, "Advances in architectural geometry 2014," 2014.

[12] H. Pottmann, M. Eigensatz, A. Vaxman, and J. Wallner, "Architectural geometry," *Computers & Graphics*, 2014.

[13] C.-H. Peng, Y.-L. Yang, F. Bao, D. Fink, D.-M. Yan, P. Wonka, and N. J. Mitra, "Computational network design from functional specifications," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 131:1–131:12, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925935

[14] F. Bao, D.-M. Yan, N. J. Mitra, and P. Wonka, "Generating and exploring good building layouts," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 122, 2013.

[15] Y.-L. Yang, J. Wang, E. Vouga, and P. Wonka, "Urban pattern: Layout design by hierarchical domain splitting," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, vol. 32, p. Article No. xx, 2013.

[16] P. Galle, "An algorithm for exhaustive generation of building floor plans," *Commun. ACM*, vol. 24, no. 12, pp. 813–825, Dec. 1981. [Online]. Available: http://doi.acm.org/10.1145/358800.358804

[17] J. Michalek and P. Papalambros, "Interactive design optimization of architectural layouts," *Engineering Optimization*, vol. 34, no. 5, pp. 485–501, 2002. [Online]. Available: http://dx.doi.org/10.1080/03052150214021

[18] H. Yi and Y. K. Yi, "Performance based architectural design optimization: Automated 3d space layout using simulated annealing," in *ASHRAE/IBPSA-USA Building Simulation Conference*, 2014.

[19] H. Liu, Y. Yang, S. AlHalawani, and N. J. Mitra, "Constraint-aware interior layout exploration for pre-cast concrete-based buildings," *The Visual Computer*, vol. 29, no. 6-8, pp. 663–673, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00371-013-0825-1

[20] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 181:1–181:12, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1882261.1866203

[21] S. A. Arvin and D. H. House, "Modeling architectural design objectives in physically based space planning," *Automation in Construction*, vol. 11, no. 2, pp. 213–225, 2002.

[22] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM Transactions on Graphics*, vol. 30, no. 4, p. 86, 2011.

[23] T. Feng, L.-F. Yu, S.-K. Yeung, K. Yin, and K. Zhou, "Crowd-driven mid-scale layout design," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 132:1–132:14, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925894

[24] L. G. Caldas and L. K. Norford, "A design optimization tool based on a genetic algorithm," *Automation in construction*, vol. 11, no. 2, pp. 173–184, 2002.

[25] A. Bassuet, D. Rife, and L. Dellatorre, "Computational and optimization design in geometric acoustics," *Building Acoustics*, vol. 21, no. 1, pp. 75–86, 2014.

[26] K. Sims, "Interactive evolution of equations for procedural models," *The Visual Computer*, vol. 9, no. 8, pp. 466–476, 1993.

[27] X. Shi and W. Yang, "Performance-driven architectural design and optimization technique from a perspective of architects," *Automation in Construction*, vol. 32, pp. 125–135, 2013.

[28] J. Felkner, E. Chatzi, and T. Kotnik, "Interactive particle swarm optimization for the architectural design of truss structures," in *Computational Intelligence for Engineering Solutions (CIES), 2013 IEEE Symposium on*. IEEE, 2013, pp. 15–22.

[29] M. Turrin, P. von Buelow, and R. Stouffs, "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 656–675, 2011.

[30] C. Ma, N. Vining, S. Lefebvre, and A. Sheffer, "Game level layout from design specification," *Computer Graphics Forum*, vol. 33, no. 2, pp. 95–104, 2014. [Online]. Available: http://dx.doi.org/10.1111/cgf.12314

[31] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Designscape: Design with interactive layout suggestions," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 2015, pp. 1221–1224.

[32] E. Brochu, T. Brochu, and N. de Freitas, "A bayesian interactive optimization approach to procedural animation design," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '10. Goslar Germany, Germany: Eurographics Association, 2010, pp. 103–112. [Online]. Available: http://dl.acm.org/citation.cfm?id=1921427.1921443

[33] N. Umetani, T. Igarashi, and N. J. Mitra, "Guided exploration of physically valid shapes for furniture design," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)*, vol. 31, no. 4, 2012.

[34] V. G. Kim, S. Chaudhuri, L. Guibas, and T. Funkhouser, "Shape2Pose: Human-Centric Shape Analysis," *Transactions on Graphics (Proc. of SIGGRAPH)*, vol. 33, no. 4, 2014.

[35] M. Harada, A. Witkin, and D. Baraff, "Interactive physically-based manipulation of discrete/continuous models," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 199–208. [Online]. Available: http://doi.acm.org/10.1145/218380.218443

[36] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, "Interactive furniture layout using interior design guidelines," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 87:1–87:10, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964982

[37] R. Akase and Y. Okada, "Web-based multiuser 3d room layout system using interactive evolutionary computation with conjoint

analysis," in *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction*, ser. VINCI '14. New York, NY, USA: ACM, 2014, pp. 178:178–178:187. [Online]. Available: http://doi.acm.org/10.1145/2636240.2636849

[38] C. D. Twigg and D. L. James, "Many-worlds browsing for control of multibody dynamics," in *Proceedings of ACM SIGGRAPH*. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1275808.1276395

[39] S. Agrawal, S. Shen, and M. van de Panne, "Diverse motions and character shapes for simulated skills," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 10, pp. 1345–1355, Oct 2014.

[40] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber, "Design galleries: A general approach to setting parameters for computer graphics and animation," in *Proceedings of ACM SIGGRAPH*, 1997, pp. 389–400. [Online]. Available: http://dx.doi.org/10.1145/258734.258887

[41] E. Hebrard, B. Hnich, B. O'Sullivan, and T. Walsh, "Finding diverse and similar solutions in constraint programming," in *AAAI*, vol. 5, 2005, pp. 372–377.

[42] R. K. Ursem, *Parallel Problem Solving from Nature — PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, ch. Diversity-Guided Evolutionary Algorithms, pp. 462–471. [Online]. Available: http://dx.doi.org/10.1007/3-540-45712-7_45

[43] B. Srivastava, T. A. Nguyen, A. Gerevini, S. Kambhampati, M. B. Do, and I. Serina, "Domain independent approaches for finding diverse plans." in *IJCAI*, 2007, pp. 2016–2022.

[44] A. Coman and H. Muñoz-Avila, "Generating diverse plans using quantitative and qualitative plan distance metrics." in *AAAI*. Citeseer, 2011, pp. 946–951.

[45] B. Hillier and J. Hanson, "The social logic of space, 1984," *Cambridge: Press syndicate of the University of Cambridge*, 1984.

[46] J. Peponis, C. Zimring, and Y. K. Choi, "Finding the building in wayfinding," *Environment and behavior*, vol. 22, no. 5, pp. 555–590, 1990.

[47] A. Turner and A. Penn, "Making isovists syntactic: isovist integration analysis," in *2nd International Symposium on Space Syntax, Brasilia*. Citeseer, 1999.

[48] G. Franz, H. Mallot, J. Wiener, and K. Neurowissenschaft, "Graph-based models of space in architecture and cognitive science-a comparative analysis," in *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics*, vol. 3038, 2005.

[49] D. Dara-Abrams, "Architecture of mind and world: How urban form influences spatial cognition," in *Proceedings of the Space Syntax and Spatial Cognition of the Workshop at Spatial Cognition, Bremen, Germany*, vol. 24, pp. 1–57.

[50] C. Davies, R. Mora, and D. Peebles, "Isovists for orientation: can space syntax help us predict directional confusion?" in *Space syntax and spatial cognition: Proceedings of the workshop held in Bremen*, vol. 2, 2006, pp. 81–92.

[51] T. Meilinger, G. Franz, and H. H. Bülthoff, "From isovists via mental representations to behaviour: first steps toward closing the causal chain," *Environment and Planning B: Planning and Design*, vol. 39, no. 1, pp. 48–62, 2012.

[52] B. Emo, C. Hoelscher, J. Wiener, and R. Dalton, "Wayfinding and spatial configuration: evidence from street corners," pp. 1–16, 2012.

[53] M. Bhatt, C. Schultz, and M. Huang, "The shape of empty space: Human-centred cognitive foundations in computing for spatial design," in *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*. IEEE, 2012, pp. 33–40.

[54] T. W. Kim and M. Fischer, "Ontology for representing building users activities in space-use analysis," *Journal of Construction Engineering and Management*, vol. 140, no. 8, p. 04014035, 2014.

[55] M. Kapadia, N. Pelechano, J. Allbeck, and N. Badler, "Virtual crowds: Steps toward behavioral realism," *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, vol. 7, no. 4, pp. 1–270, 2015.

[56] A. Turner, "A program to perform visibility graph analysis," in *Proceedings of the 3rd Space Syntax Symposium, Atlanta, University of Michigan*, 2001, pp. 31–1.

[57] W. Hillier, J. Hanson, and J. Peponis, "Syntactic analysis of settlements," *Architecture et comportement/Architecture and Behaviour*, vol. 3, no. 3, pp. 217–231, 1987.

[58] B. Jiang, C. Claramunt, and B. Klarqvist, "Integration of space syntax into gis for modelling urban spaces," *International Journal of Applied Earth Observation and Geoinformation*, vol. 2, no. 3, pp. 161–171, 2000.

[59] J. Desyllas and E. Duxbury, "Axial maps and visibility graph analysis: A comparison of their methodology and use in models of urban pedestrian movement," in *3rd International Space Syntax Symposium*, 2001, pp. 27.1–27.13.

[60] ——, "Axial maps and visibility graph analysis," in *Proceedings, 3rd International Space Syntax Symposium*, vol. 27. Georgia Institute of Technology Atlanta, 2001, pp. 21–13.

[61] C. Hölscher, M. Brösamle, and G. Vrachliotis, "Challenges in multilevel wayfinding: A case study with the space syntax technique," *Environment and Planning B: Planning and Design*, vol. 39, no. 1, pp. 63–82, 2012.

[62] A. Turner, M. Doxa, D. O'sullivan, and A. Penn, "From isovists to visibility graphs: a methodology for the analysis of architectural space," *Environment and Planning B: Planning and design*, vol. 28, no. 1, pp. 103–121, 2001.

[63] M. Usman, D. Schaumann, B. Haworth, G. Berseth, M. Kapadia, and P. Faloutsos, "Interactive spatial analytics for human-aware building design," in *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, ser. MIG '18. New York, NY, USA: ACM, 2018, pp. 13:1–13:12. [Online]. Available: http://doi.acm.org/10.1145/3274247.3274503

[64] J. Neetil and P. O. de Mendez, *Sparsity: Graphs, Structures, and Algorithms*. Springer Publishing Company, Incorporated, 2012.

[65] C. Hölscher, T. Meilinger, G. Vrachliotis, M. Brösamle, and M. Knauff, "Finding the way inside: Linking architectural design analysis and cognitive processes," in *Spatial Cognition IV. Reasoning, Action, Interaction*. Springer, 2004, pp. 1–23.

[66] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation," in *IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.

[67] A.-T. Nguyen, S. Reiter, and P. Rigo, "A review on simulation-based optimization methods applied to building performance analysis," *Applied Energy*, vol. 113, no. Supplement C, pp. 1043 – 1058, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261913007058

[68] J. Brooke, "Sus: a retrospective," *Journal of usability studies*, vol. 8, no. 2, pp. 29–40, 2013.

[69] J. Sauro and J. R. Lewis, "When designing usability questionnaires, does it hurt to be positive?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2215–2224.

[70] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

[71] M. Usman, D. Schaumann, B. Haworth, M. Kapadia, and P. Faloutsos, "Joint Parametric Modeling of Buildings and Crowds for Human-Centric Simulation and Analysis," in *Proceedings of the Computer-Aided Architectural Design Futures.*, Daejong, Korea, 2019.