

Article

Architectural Design Exploration Using Generative Design: Framework Development and Case Study of a Residential Block

Jani Mukkavaara *  and Marcus Sandberg 

Construction Management and Building Technology, Department of Civil,
Environmental and Natural Resources Engineering, Luleå University of Technology, 971 87 Luleå, Sweden;
marcus.sandberg@ltu.se

* Correspondence: jani.mukkavaara@ltu.se

Received: 14 October 2020; Accepted: 2 November 2020; Published: 4 November 2020



Abstract: The use of generative design has been suggested to be a novel approach that allows designers to take advantage of computers' computational capabilities in the exploration of design alternatives. However, the field is still sparsely explored. Therefore, this study aimed to investigate the potential use of generative design in an architectural design context. A framework was iteratively developed alongside a prototype, which was eventually demonstrated in a case study to evaluate its applicability. The development of a residential block in the northern parts of Sweden served as the case. The findings of this study further highlight the potential of generative design and its promise in an architectural context. Compared to previous studies, the presented framework is open to other generative algorithms than mainly genetic algorithms and other evaluation models than, for instance, energy performance models. The paper also presents a general technical view on the functionality of the generative design system, as well as elaborating on how to explore the solution space in a top-down fashion. This paper moves the field of generative design further by presenting a generic framework for architectural design exploration. Future research needs to focus on detailing how generative design should be applied and when in the design process.

Keywords: architectural design; computer-based; design exploration; framework; generative design; residential block; solution space; Swedish case study

1. Introduction

The challenge for building designers is increasing as the act of balancing multiple requirements and needs imposed by clients and regulations on cost, space, esthetics, and sustainability becomes harder. Meeting these requirements and needs can require novel design solutions and expanding the design's solution space beyond that of previous designs. This process can involve both exploration and exploitation, depending on the project situation and the capabilities of the design team. However, time and resource limitations can restrict the opportunities for exploration [1].

In exploration, the interest lies in probing an extensive set of solutions to find interesting regions within a design's solution space. When a region is identified it can be narrowed down to find the best solutions subject to a set of criteria. The act of creating these novel solutions can, however, be resource intensive. To facilitate a design process that ends up with strong design alternatives, an efficient generation and representation of a wide range of feasible solutions is needed [2]. One avenue that has received research attention within this area is the application of optimization approaches to reach near-optimum solutions for building designs regarding cost, energy efficiency, or carbon emissions (e.g., [3–6]). The goal of optimization is to find an optimal solution (or a set of optimal solutions,

often referred to as Pareto solutions, in the case of multi-objective optimization) to a design problem. The application of optimization thus weighs more heavily into exploitation rather than exploration, which means there is a likelihood of ending up in a solution similar to already-existing solutions. Application of methods like multi-objective optimization has previously shown its ability to offer solutions outperforming the initial designs of concrete problems, such as life-cycle energy reductions in building design [7]. The application of optimization thus has great potential at the later stages of design, when the design requirements are more defined.

Previous research has, however, pointed out a need to also support the exploration aspect in the earlier stages of design through generative design approaches (e.g., [2,8,9]). Generative design is a term used to describe a wide range of methods with the aim of assisting product designers by using computational capabilities to generate feasible solutions and one of generative design's primary objectives being the ability to explore larger solution spaces [8]. The goal of generative design is to provide novel findings, rather than revealing what is already known, thus expanding designers' reference frames. Despite its heavy emphasis on utilizing computational capabilities, and in contrast to fully autonomous methods, generative design is described as using computers as a 'collaborative partner' [10]. Its usefulness is argued to be primarily found during the conceptual stages of design, where the design is still under formulation [2]. With these underlying characteristics, generative design is in this work defined as the use of computer-based methods in interactions with designers to generate feasible solutions that facilitate exploration of a design's solution space with the goal of finding novel solutions.

There are multiple ways that a generative design approach can be realized. For example, Singh and Gu [8] presented a review of five generative design techniques (L-systems, cellular automata, genetic algorithms, swarm intelligence, and shape grammars). They identified potential uses of each technique given a design problem's characteristics and highlighted each technique's benefits and challenges. Based on their study, they argued that no technique could match all design problems' needs, and that those needs might change during the process itself. It was, therefore, suggested that a generative design system needs to be flexible and provide agency to its user to select techniques used both initially but also progressively during a generative design process. Given this, generative design has a large scope of potential implementations and applications. Several frameworks, methods, and tools have been presented within the generative design research of which a set of relevant research publications are herein discussed. First, an example from the manufacturing industry is mentioned, followed by a structural engineering example, then several architectural examples are presented.

In a study by Krish [2], the outline for a generative design method driven by a history-based parametric CAD-system and a random sampling technique to generate design alternatives was suggested. This method was then exemplified using a consumer product (an MP3 player). From their comparison with genetic algorithms, they argued that the use of a random sampling-based generative design approach is more suitable for conceptual design problems because of its ease of use, ability to use non-quantifiable criteria (e.g., aesthetics), and problem maturity. Shea et al. [10] presented a study where they integrated a structural design system (called eifForm) with a modeling system (called Generative Components) to enable exploration of design alternatives in a performance-driven generative design process. Their study provided a practical example for the application of generative design on structural components and highlighted the importance of understanding how geometric relationships and modifications affect generative methods.

Nagy et al. [11] described a workflow for generative design applied to architectural indoor space planning, where six performance metrics (adjacency preference, work style preference, buzz, productivity, daylight, views to outside) were subjected to a multi-objective genetic algorithm to generate design alternatives. Although their findings were encouraging, they described a need for more flexibility in their methods so that additional parameters could be exposed and controlled by multi-objective genetic algorithms. They also highlighted that the output from a generative design workflow should not be thought of as final, but as a baseline from where further analysis and development is needed to finalize

the design. Abrishami et al. [12] integrated a generative system with a building information modeling (BIM) environment to support conceptual design in an architecture, engineering, and construction (AEC) context by providing techniques for design solution generation and exploration. Their proposed framework was also recently developed into a prototype (see [13]), with an emphasis on embedding information into generated solutions. Their findings showed promising results in the approach's ability to support creativity and process information. However, they suggested further studies to investigate a broader range of design settings. Another example of a BIM-based application is found in Salimzadeh et al. [14], where they proposed a method that uses a BIM-based process to generate layouts for photovoltaic (PV) modules and calculate the generated solution's performance. Their proposal leverages object data in BIM models such that surface-specific conditions can be incorporated in the generation of PV module layouts, showing how BIM data can be incorporated as input to a generative design approach. A related concept can be found in Hamidavi et al. [15], where they investigated the link between architects and structural engineers by proposing an automatic generation and optimization of structural models based on input data gathered from architectural models. Their study provides valuable insights into the potential usage of a generative design approach by linking models from multiple disciplines and how constraints derived from traditionally labor-intensive tasks can be alleviated by performing automatic generation based on those links. Gerber and Lin [9] combined parametric modeling with multi-objective optimization to support the generation and evaluation of building project scenarios using energy simulation and financial models. Their findings suggested that having a high degree of automation in the evaluation of solutions was beneficial to the users by offering users instant feedback on design alternatives, as they tended to focus more on multiple objectives simultaneously, rather than on one objective at a time. Nagy et al. [16] sought to investigate the potential of generative design on an urban scale. In their study, they demonstrated the application of an optimization-based generative design process for residential neighborhood layouts in which optimization of profit and energy generation from solar panels was performed. They concluded that a generative design process has the potential to reveal insights on tradeoffs between design goals, and that those insights can lead to better-informed designs. They also highlight that generative design on an urban scale needs further research, including studies looking at the complexity of design at the urban scale through the inclusion of additional design metrics.

Research Gap, Aim, and Method

Despite the growth and increase in popularity of generative design approaches and the potential increasingly understood, the field has nevertheless been sparsely explored. Work in the field has been described by some as rudimentary [2] or in its infancy [12]. Few studies also exist within an architectural design context, whose implications can differ from those of other domains. The existing studies within architectural design present frameworks with accompanying prototypes. Some frameworks are more focused on genetic algorithms [11,16], energy performance [9], prototype development [13], or the design process [12], for example. Therefore, there is a need for frameworks focusing on the general and technical workflow of the generative architectural design system. There is also a need for a better understanding of how the design's solution space should be represented to enable different types of exploration (e.g., bottom-up and top-down [8]). Thus, further studies are needed that explore the applications of generative design in the architectural design context, and that provide additional findings that contribute to the formation of the generative design knowledge domain.

The aim of this study is thus to investigate the potential use of generative design in an architectural design context. This is done by iteratively developing a framework together with a functioning prototype that is eventually demonstrated in a case study to evaluate its applicability in a case of designing a residential block. The research design was based on a methodology from design science research by Peffers et al. [17], where a problem drives a cycle of objectives (development, demonstration, and evaluation) from which suggestions are communicated by evaluating and reflecting on the results against the existing theory (see Figure 1).

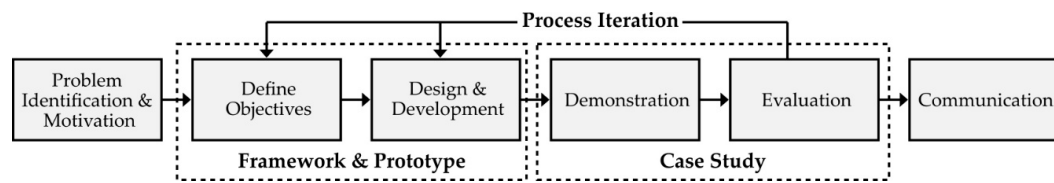


Figure 1. Overview of the research design in this study (based on the methodology described in [17]).

The iterative development of a framework and prototype was deemed suitable, as it allows for insights that are gathered during testing and demonstration to be piped back to the framework to improve its design. Furthermore, a case study approach was used for its potential in examining a phenomenon in a natural setting [18], thus enabling the evaluation of the developed framework. Collectively, the exploratory approach of using a case coupled with the design and development of a framework and prototypes was chosen as the research design.

Three primary activities were iterated during the research process: framework development, prototype development, and case demonstration. The framework development was the initial activity, where abstraction of a system was described based on previous knowledge gathered from previous research and the authors' own experiences. The use of previous research helped in the design and development of the framework by contributing existing theories (e.g., regarding the selection of generative methods [8] or system architecture [19]) to the framework's formulation. The framework provided an outline of the generative design process, which in the subsequent activity was developed into a functioning prototype. The role of the prototype was to provide a platform from where a demonstration of the framework could be performed, and to assist in identifying potential obstacles in its technical implementation. The third activity dealt with demonstrating the use of the developed framework and prototype in a case study. When subject to a problem in a real-world context, knowledge and experiences can be gathered and used to refine and provide meaningful insights for the further development of the methods under investigation. These three activities were iterated throughout the research process, where knowledge gathered was used to revise and further refine the developed framework and prototype. The framework is described in Section 2, and the prototype and case study are described in Section 3.

2. Generative Design Framework

The study in this paper first set out to develop a framework for generative design that could later be applied to the residential block case. The developed framework supports the application of a generator to derive solution sets for a design problem bound by a solution space. Relevant models are created for each solution in the solution set, and evaluations are executed on those models to derive metrics that describe each solution. This serves as input to the exploration approach where solutions and their metrics are presented and subjected to preference management, and the filtering and selection of interesting solutions either result in a final set of feasible solutions or as input for a solution space modification in an exploration loop. Figure 2 shows an overview of the framework, and the following sections describe it in more detail.

2.1. Solution Space

The solution space is the first point of contact in our framework. A solution space provides the boundaries, or what Krish [2] calls the 'performance envelopes' within which feasible solutions exist. The term feasible solution here refers to a solution that lies within the boundaries of the solution space and fulfills all defined constraints. Thus, a feasible solution is one that could be chosen as the final solution of a generative design process. The solution space is there to capture design logic that is defined before the actual generation process, a technique also used in parametric modeling to facilitate rapid changes and design exploration through manipulation [10]. The objective of the solution space is to narrow down the pool of alternative solutions, with the goal being to provide the exploration with

a meaningful set of feasible solutions. By iteratively modifying the solution space, the exploration can progress towards a direction that provides an increasingly meaningful set of feasible solutions.

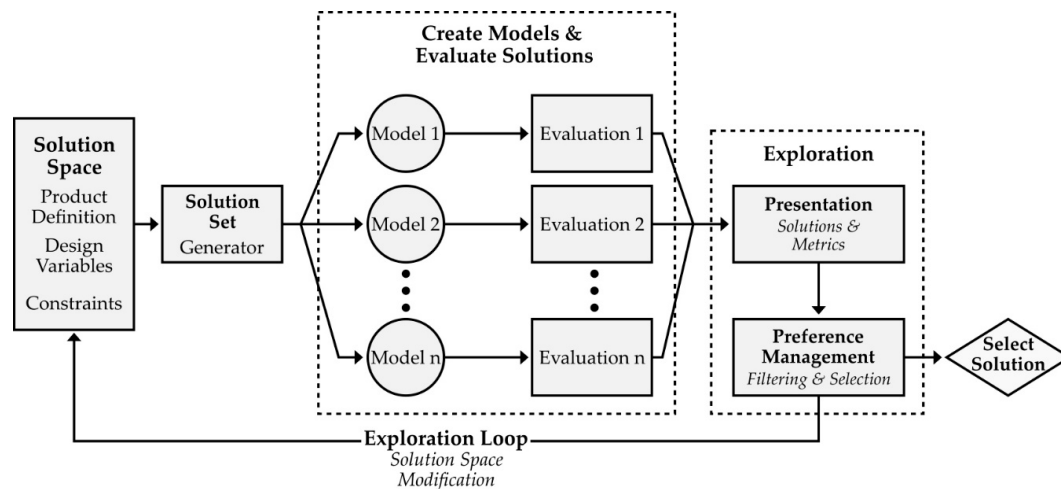


Figure 2. Overview of the developed framework for generative design.

The solution space is defined by its three components: the product definition, design variables, and constraints. At the base of the solution space is the product definition. The term ‘product’ is here used as a term to define the entity that is subject to the generative design approach. The product can represent the results of an entire construction project or a subsection of it. A product can, for example, be a load-bearing component, floor plan, building (or sections of it), building site (with one or more buildings), or a city block. The product definition is there to describe the overall product (e.g., through a geometric model) for the design variables and constraints to interact with and to define constants and rules that apply in the generation of a feasible solution.

To establish a set of alternatives to explore, design variables are used to define variations in the solutions. Each design variable describes either a continuous range of values bound by lower and upper limits, or a discrete set of values. Design variables can affect both the physical representation (e.g., dimensions) of a product but also its properties (e.g., thermal properties through material selection). Examples of design variables include the length and width of a building, the slope of a roof, the thickness of a slab, material selection in a wall, and so on. By changing the design variables different solutions can be created, and it is these design variables that the generator will choose from in order to generate the solution set for the current iteration. Through design variables, boundaries are set for what constitutes a feasible solution; however, it is not always achievable to express these boundaries solely using the design variables. As a complement, constraints can define additional boundaries to the solution space. Examples of constraints include the total habitable floor area for a building that can vary its footprint and number of floors, or the total thickness of a wall when each layer’s thickness in a wall can be varied individually. Furthermore, constraints can be used to define the boundaries of a solution’s metrics to, for example, meet regulations regarding energy consumption.

Each design variable and each constraint that is used defines a boundary for the solution space, and the final solution space is the areas where each of the boundaries overlap. Because of the inherent effects of defining design variables and constraints on the solution space, careful consideration should be given to their choices, as they can affect the selection of generation methods [8] and the outcome of the generative design process. Similar to parametric modeling, by carefully choosing design variables for an exploration, a more accurate solution space can be defined that is expected to lead to more meaningful conclusions [20].

2.2. Solution Set and Generator

In order to meet generative design's primary objective of exploring extensive solution spaces [8] and discovering novel solutions [10], there need to be systems in place that enable the generation of solutions. Thus, the generator is at the core of the generative design approach—it provides the 'generative' aspect of generative design. The output of the generator is a set of feasible solutions that are available for exploration.

Here, we define the generator as a method that can generate solutions based on the definitions and boundaries of the solution space, with its goal being to autonomously generate a set of solutions. There are multiple different methods that can be chosen for implementing the generator (e.g., see [8]). A commonly leveraged method found in previous research on generative design is one based on genetic algorithms (e.g., [9,11,16]). A genetic algorithm approach is driven by a set of biologically inspired processes (e.g., mutation, crossover, and selection), where each solution is evaluated by a fitness function (a function that numerically depicts the performance of a solution based on a given problem's objectives). In each iteration of a genetic algorithm approach, the biologically inspired processes are used to generate a solution set (called a 'population') that with each iteration increasingly improves the fitness of its solutions. Given the focus of genetic algorithms on minimizing or maximizing different objectives (e.g., minimizing cost), genetic algorithms' presence is typically seen in optimization problems, and has been shown to have potential for design problems with distinct and measurable objectives (e.g., [21]).

Despite the existence of various intelligent and advanced methods (e.g., genetic algorithms), it can also be argued that some of the methods pose a hindrance because of knowledge requirements, implementation difficulties, or time-constraints—all factors that should also be taken into consideration. For example, setting up variables and objectives is a crucial activity in genetic algorithms that can require substantial skills and knowledge [2,8]. As such, simpler methods like random sampling can offer a suitable approach for conceptual design problems that are focused on exploration [2]. A random sampling approach typically relies on pseudo-random number generators to generate solution sets by selecting solutions within the solution space at random locations. Using random sampling poses few restrictions on a problem's maturity and criteria, as random sampling is not driven by well-defined measurable objectives and is arguably easier to set up and use (in contrast to genetic algorithms, for example). This also means that the responsibility of guiding the exploration's direction is up to the user (in contrast to, for example, genetic algorithms), which can facilitate the inclusion of qualitative assessments of solutions (e.g., esthetics).

As Singh and Gu [8] also concluded in their study, each method has its benefits and drawbacks, and the method chosen should be based on problem characteristics and objectives. The framework that was developed for this study is focused on supporting those generator methods that interact with an independently defined solution space (i.e., the solution space is not necessarily tightly integrated with the generator algorithm itself), and areas where solution metrics are derived for each solution to support preference management. Thus, the framework is suitable for use of both genetic algorithms and random sampling.

2.3. Create Models and Evaluate Solutions

The generated solutions (i.e., the solution set), together with the product definition, design variables, and constraints, only provide a general representation of a solution. To facilitate a user exploration of solutions, each solution needs to be passed through a set of evaluations that are used to derive metrics for each solution. A metric is a measurable attribute of a solution and is chosen based on the design problem and its objectives. For example, a metric can be the energy performance of a building, a building's habitable area, or the weight of a component. By using multiple metrics, a diverse view on a solution can be presented, assisting the exploration stage with relevant data. This also enables a transition into the inclusion of more performance-driven aspects into the design process, a theme that is gaining more interest in architectural design [22,23].

The processing of evaluating solutions should be performed with a high level of automation to best facilitate the exploration of larger solution sets with multiple performance metrics [9]. To support this, we build on a principle described by Sandberg et al. [19], where a general representation is transformed into different models to enable evaluation. Each model contains the specific data needed for its purpose, and models are used to handle different representations of a solution so that it can be analyzed, evaluated, and visualized according to the requirements and objectives of the design problem and its metrics. Examples of models include models of energy simulations, quantity takeoffs, sunlight studies, and visualizations, among others. Depending on the model and what evaluation is required, the evaluation can include calculations, simulations, analyses, or other processes deemed necessary. The evaluations are an essential part of providing input to the exploration approach in the form of metrics and potential sources of visualization of each solution, as this are the information that will guide the generator methods' and users' decisions.

2.4. Exploration Approach

Following the model creation and solution evaluation in our generative design framework is the exploration approach itself. Because the generated solution set can be considerable in size, and in order to support users in the identification of potentially novel solutions, the generated solution set needs to be explored [2,10]. The objective of the exploration is to probe a set of solutions to identify interesting regions of solutions. The outcome should either be a solution or a set of solutions deemed satisfactory to end the generative design process, or a set of modifiers sent back to the solution space and generator to generate a new set of solutions. This consists of two stages: presentation and preference management.

In the presentation stage, the solutions and their metrics are presented to the user for their evaluation. Here, visualization plays an important role, as it allows for the transformation of data into, as described by Chien and Flemming [24], "concrete objects that can be seen". The visualization aspect is especially important in an architectural design context as it central to its practice [25], and the representation of solutions should also be given consideration in a generative design approach. The role of the presentation is to swiftly and clearly relay relevant information to the users about each solution and its metrics in such a way that the users are assisted in making well-informed decisions. This entails a combination of 2D or 3D visualizations of a product's geometry (e.g., through images, renderings, or models) and graphs and tables that describe the product's attributes and properties.

After users have been given the presentation of each solution in the solution set, the next stage is to manage their preferences. This stage builds on the notion that generative design approaches should not be thought of as autonomous solutions, but rather as 'collaborative partners' [10]. In our generative design approach, some preferences are provided a priori at the definition of the solution space through design variables and constraints. However, as the direction of the exploration is partially dependent on what knowledge and experience a user receives [26,27], these preferences need to be modified accordingly. As such, the preference management here is responsible for collecting the progressive articulation of user preferences, so that the users can navigate the solution space by continuously modifying its composition [2]. Combined, this means that there need to be support methods available that allow users to interactively explore the solutions through selection and filtering, to evaluate their fit to the current problem's objectives, and to apply preferences that narrow down a solution space into smaller regions. This entire process, from solution space definition to exploration, is iterated until a satisfactory set of solutions is found and selected. That solution set can then be brought forward in the design process to be further developed into the final design of the product.

3. Demonstration of the Framework

3.1. Case Description

The case study performed in this study was a part of a larger research project that investigated aspects of sustainability in the development of new residential blocks. This study was a part of that

project, with the purpose of studying novel methods that can support early design processes. The case in question for this study was of a residential block in Kiruna, in the northern part of Sweden (see Figure 3).



Figure 3. Location of the residential block in relation to adjacent urban elements.

The residential block is a part of the new city center in Kiruna that is being developed. Data and illustrations for the residential block were gathered from publicly available planning documents from the municipality for the area containing the block and surrounding areas.

The block is of a rectangle shape with the dimensions 62 m × 57 m. On all four sides of the block are roads that form the city's infrastructure. Three sides of the block (south, east, and west) are adjacent to other building blocks in the area. The last side of the block (north) is adjacent to a large public park. The case study was performed together with a public property owner who was interested in developing the block with apartment buildings. The case was at a very early stage of development during the time of this study, and focus was still on conceptualization and potential ideas for the residential block.

3.2. Prototype

During the development of the framework, a prototype was iteratively developed for the case study to evaluate the use of generative design. The prototype consisted of three primary sections: one for defining the solution space and generating solution sets, one for creating models and evaluating them, and one for presenting the results and gathering preferences.

The first and last stages were both implemented using a set of custom scripts created by the authors in the programming language Python within a Jupyter Notebook environment. This choice was made as it allowed for a high degree of flexibility and ease of use regarding data management, presentation, and visualization, which was deemed helpful because of the exploratory nature of the study. Presentation of solutions was performed by linking data from the Python scripts to the three.js library for interactive 3D visualizations, and all graphs were created using the plotly library. Data from the Python scripts in the Jupyter Notebooks were sent using JavaScript Object Notation (JSON) files to the CAD software Rhino for the generation of models and the subsequent evaluation of each solution. Within Rhino, the visual programming extension Grasshopper was used along with the Ladybug tools package. This combination was chosen for its abilities and flexibility regarding generating and working with geometry, and its ability to offer easy access to various building performance evaluation tools. Within Grasshopper another custom Python script was used to receive and send JSON data for each specific solution and transform

the solutions into the models required for the evaluations. A set of Python scripts, built-in functions, and tools from the Ladybug tools package were responsible for all computation of the metrics.

3.3. Solution Space Definition

The first stage in the case study was to analyze the residential block and set the boundaries for the solution space in the generative design approach through the product definition, design variables, and constraints. The primary guidance for these activities was from documents from the municipality in Kiruna that regulates the area, combined with data gathered from the case company during a set of meetings carried out during the project's duration. The data gathered from the case company included their intent for the residential block and general guidelines for the project (e.g., habitable area, types of facilities, building typologies, etc.).

It was decided that a site composition of between one and four buildings should be used. This range was chosen to not only provide a difference in an individual building's design, but also the block's design as a whole, with the reasoning being that different site compositions (number of buildings and their positions) together with different building designs could provide potentially interesting solutions. The regulations from the municipality dictated that buildings should, to a great extent, be placed along the edges of the residential block. As such, each of the four potential buildings was assigned one of the edges of the rectangular building block. Each building was then given a set of design variables that could be individually controlled (see Table 1).

Table 1. List of design variables and their alternatives for each building that were used in the case study of the residential area, where $n \in [1,2,3,4]$ corresponds to a specific building.

Design Variable	Alternatives
Include Building _n ¹	True, False
Building _n Length	0–100% of the site's edge
Building _n Position	0–100% offset ²
Building _n Width (Start)	8–15 m
Building _n Width (End)	8–15 m
Building _n Height (Start)	2–4 floors or 2–7 floors ³
Building _n Height (End)	2–4 floors or 2–7 floors ³

¹ The notation n , where $n \in [1,2,3,4]$, corresponds to one of the four possible buildings in the residential area.

² The position of a building, or its offset along the block's edge, is dependent on the available edge length, which depends on the building's length. ³ On different parts of the site, different building heights are allowed. For the northern part of the block the smaller span is active, otherwise the greater span is active.

The ranges for a building's widths were chosen based on residential building typologies in the area and projects that the case company had previously been developing. For the heights of each building, regulations from the municipality were used where the maximum allowed height was specified. Here, the block had two regulations for different parts of it, hence the lower range for building 3 seen in Table 1. The buildings' lengths were defined as a relative value based on the length of the block's edge where that building was located. Lastly, each building's position was a value relative to how much it could move along the block's edge whilst still being inside the boundaries of the block.

Based on the design variables, sets of constraints and rules were defined that controlled the transformation from design variables to a feasible solution. This led to the definition of a process in which a building is defined (see Figure 4).

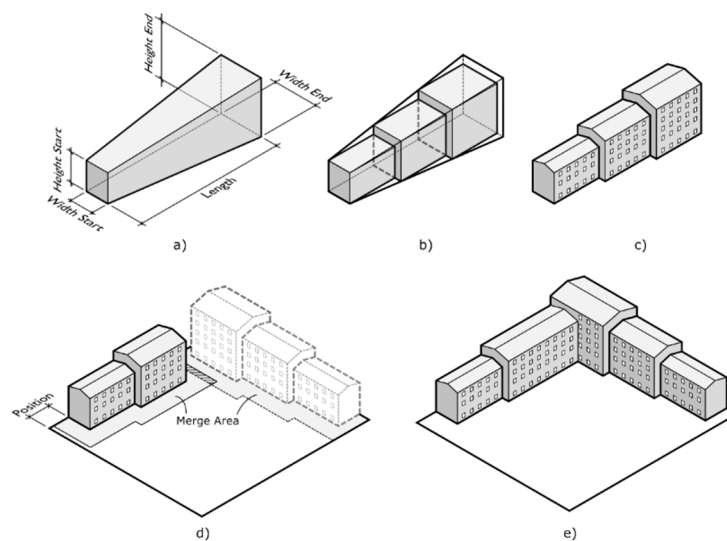


Figure 4. Outline of the building generation technique used in the case study, where (a) shows the definition of the building's bounding shape, (b) shows the fit of cuboids into the bounding shape, (c) shows the initially generated building(s), (d) shows the positioning of the building(s) and merge areas, and (e) shows a scenario where two adjacent buildings are merged.

The first step in that process is to define a bounding shape for the building that represents the geometrical boundaries within which the building resides. The bounding shape is defined using its lengths, its width at each end, and its height at each end. Next, a set of cuboids are fit into the bounding shape, where each cuboid represents a section of the building. A rule was defined where the length of the building defines how many segments it will be divided into, and thus also how many cuboids are fit into the bounding shape. This rule was defined as the building's length divided by 20 m, a value that was chosen based on empirical testing until a suitable division was found for the case in question. For each cuboid, a simple building section is then generated, and each building is positioned along its edge in the block. Next, a merge area is generated around each building. The purpose of this area is to identify buildings (or sections of buildings) that are in proximity to each other and are thus subject to being merged. A threshold value was chosen for the size of the merge areas that dictated that buildings closer than 6 m to each other be merged into a single building. The merging process itself extended the closest building sections in such a way that they intersected with each other.

3.4. Generation of Solution Sets

To generate the solutions sets, this case study applied a random sampling technique as the generator. The primary reason behind this choice was the emphasis on exploration, rather than exploitation. Although genetic algorithms have been a popular suggestion for the generation of solutions in generative design approaches in the past, they tend to be more applicable when concrete objectives have been identified, and when exploitation has a higher priority. Furthermore, it was determined that the simplicity of a random sampling technique was suitable as it shifted focus away from the intricacies of specific algorithms towards a general approach of applying generative design to a design problem instead.

The random sampling generation was implemented by using a pseudo-random number generator to select different combinations of design variable values, where each combination represented a potential solution. For each iteration of the generative design process, a set of 50 random solutions were automatically generated based on the definitions in the solution space. This size was chosen for the solution sets with the intent of providing a broad variety of solutions, whilst simultaneously keeping the number of solutions relatively low to make user exploration more manageable.

3.5. Models and Their Evaluation

The metrics chosen to evaluate each solution in the case study were sunlight hours, view, habitable area, disposable area, and variation. The metrics were chosen with the explorative nature of this study in mind, and with directions and interests of the case company considered. The sunlight hours metric was chosen primarily due to the northern location of the building site, as sunlight is limited during part of the year (around winter). Therefore, it was deemed interesting to evaluate how different solutions performed in contrast with the number of hours of sunlight that the block was exposed to. The sunlight hours evaluation used a geometric model containing the block, surrounding buildings, and a geometric model of the current solution itself (see Figure 5).

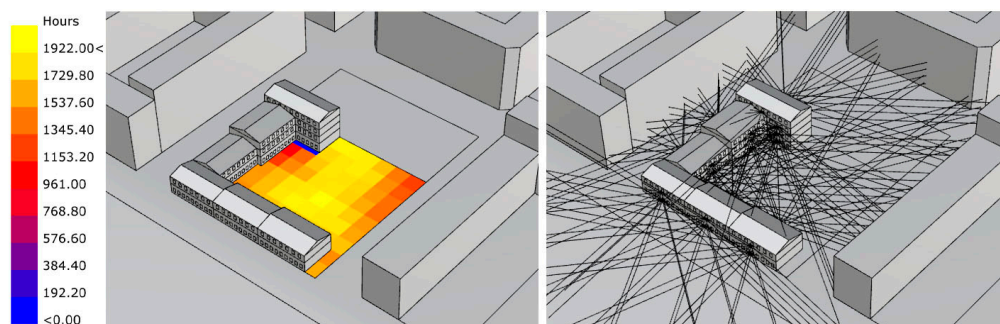


Figure 5. Example of models and evaluation of sunlight hours (left) and views (right) for a solution.

The block was divided into cells of 8 m × 8 m, and the total amount of potential hours of sunlight that each cell received during an entire year was calculated. This cell size was chosen as a middle ground for balancing calculation time (a smaller cell size requires more extensive calculations, i.e., more time) and accuracy.

The same model as for sunlight hours was used for the calculation of views. The view calculation considered the potential views that could be had from each generated window in a given solution. This evaluated the ‘openness’ of a solution, where a more open solution was rewarded with a higher score, as a longer view range was deemed more desirable. For the calculations, a coarse isovist analysis was performed, with its origin points at the center of each window (see Figure 5).

The view range of each ray was determined by calculating the distance from the origin point (i.e., the center of a window) to the point where the isovist ray intersected either the building itself or any of the surrounding buildings. A value of 100 m was set as the maximum distance at which an intersection would be searched for, due to the surrounding buildings only occupying the sites closest to the building site and the far distance being empty in the model. The average value of the view range was used as the metric for a solution.

For the two metrics related to areas, a simplified geometric model was used, as it only needed to quantify the floor areas of a solution. The first of these area-based metrics was the habitable floor area. The case company had an initial target of 3500 m² for the total habitable floor area for the residential block in question. This was derived from a distribution of apartments and apartment sizes combined with a utilization factor of 0.8 to account for service areas, stairwells, elevators, and so on. For the case demonstration, it could have been used as a constraint where each solution had to target the set habitable floor area; however, it was decided that the generative design approach would facilitate a more exploratory approach if the solution space was not bound by a floor area constraint, as that would limit the variation in solution alternatives greatly. As a result, the habitable floor area was used as a metric, where a value closer to their target of 3500 m² had a higher score than values further away from it. Besides using the habitable floor area as a metric, the remaining area available on the site (hereafter referred to as the ‘disposable area’) was also computed and used as a metric. This was carried out because different solutions (in terms of the number of buildings, their sizes, and their compositions) took up different amounts of space on the block. During the study, it was conveyed

by the case company that the block was going to struggle with the available area on the site after the buildings were placed. The block should, preferably, contain facilities for waste management, car parking, and outdoor recreation (such as playgrounds, grass surfaces, etc.). Thus, a metric related to the disposable space on the block was deemed a potentially useful indicator to include.

The last metric included was that of building variation. It was stated in the regulations from the municipality that buildings (and especially larger structures) constructed on the site had to have variations to some extent to break up their volumes. The reasoning in the regulations was that large monoliths were something to strive away from, and that building designs should offer variation to the surrounding environment. Given this, a metric for variation was calculated using a mathematically defined model that used the design variables for a solution as input. The greater the variation in the design variables, the higher the score for the metric.

3.6. Exploration of Feasible Solutions

Each solution in the set was presented together with their metrics using 3D models, graphs, and tables (see Figure 6). All the presented solutions were generated using the parameters described in Table 1 and following the process outlined in Figure 4.

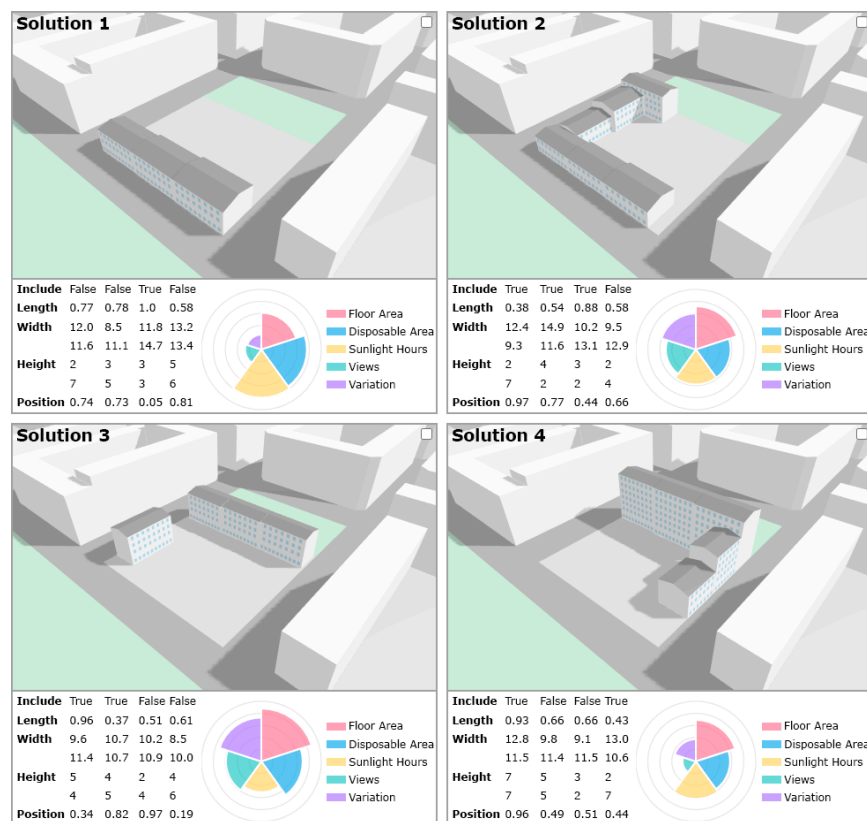


Figure 6. Examples of individual solutions from a generated solution set and their metrics, including an interactive 3D visualization, a table for a solution's design variables, and a polar chart for a solution's metrics.

For each solution, an interactive 3D model was used to visualize the residential block, its surroundings, and the suggested building composition on it. The interactive 3D model allowed the user to zoom, pan, and rotate the view so that they could observe the solution from any angle they deemed interesting. Accompanying the 3D visualization was a table describing the design variable values for the solution that highlighted the details of the current solution (e.g., the heights of each building). Lastly, the metrics of each solution were displayed using a simple polar chart, where each metric was normalized concerning the other solutions in the set, thus showing their relative difference.

Accompanying the presentation of individual solutions, a presentation of the entire solution set was also included (see Figure 7).

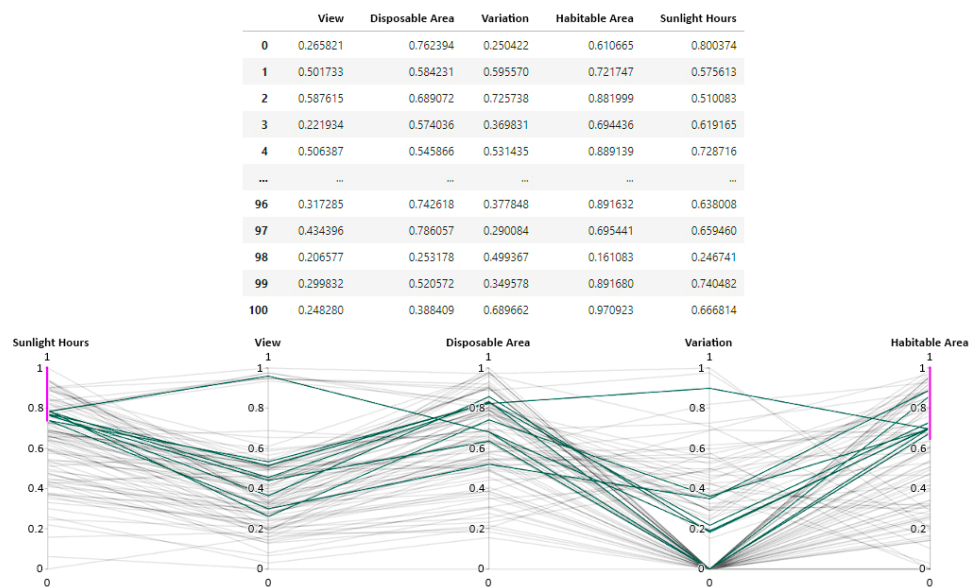


Figure 7. Example of a solution set and its metrics, showing a table of solution metrics and a parallel coordinate chart of solution metrics.

This presentation showed a table for all solutions, and a parallel coordinates chart was used to show the distribution of values for each design variable in the solution set. This allowed for an overview of the current state of the solution space.

For entering user preferences, individual solutions that were deemed interesting or relevant could be selected using a toggle in the presentation view seen in Figure 5. When entering the next iteration of the generative design process, a new solution space was defined using the selected solutions as references. For each selected solution, a region in the solution space was defined by only allowing design variable values that were within proximity of those of the selected solution. A threshold was used for each value to create new design variable ranges (e.g., $\pm 10\%$). Since multiple different solutions could be selected, two different approaches were tested (see Figure 8): the first allowed different ‘islands’ to be defined in the modified solution space (up to one per selected solution), while in the second the regions only modified the outer boundaries of the solution space.

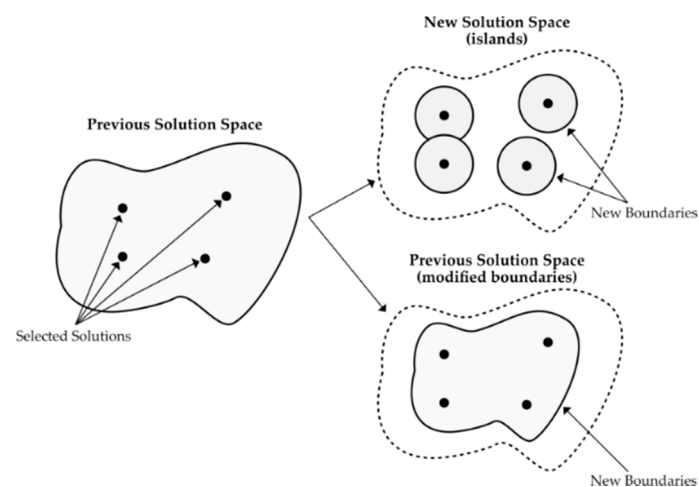


Figure 8. Two approaches to modifying the solution space based on user preferences: creating islands surrounding each selected solution, or creating a modified boundary based on all selected solution.

The first of these approaches was able to more quickly narrow down the solution space and reduce the exploration, as it restricted the boundaries into multiple smaller regions (whose sizes were determined by the thresholds for each design variable). On the other hand, the second approach retained a greater solution space, especially if the user selected multiple solutions that were considerably different in composition to each other. The drawback, however, is that this can require additional iterations in order to reduce the solution space sufficiently enough from its initial size to provide meaningful insights.

Regardless of the approach, preferences from the user were gathered and the solution space could be modified iteratively multiple times. This is what allowed for an exploratory approach, where the solution space was navigated in search of interesting solutions. This entire process of iterating between generating and evaluating solutions and continuously modifying the solution space was performed until a satisfactory set of solutions was identified. Selected solutions could be further developed in more detail in the latter parts of a design process.

4. Discussion

4.1. Research Contribution

Using computational methods to facilitate design exploration can support the iterative design process of finding novel solutions to requirements and needs imposed by clients and regulations. In this realm, generative design has been suggested as a ‘collaborative partner’ [10] that takes advantage of computers’ computational capabilities to assist in the exploration of design alternatives. Although interest in the application of generative design has received increased attention, the field has been sparsely explored, especially in the architectural design context. This paper contributes by developing a framework for generative design and evaluating its potential through a demonstration using in the real-world case of residential block development. The presented framework differs from earlier presented frameworks in some ways: firstly, by not limiting the generative design system to genetic algorithms (as in Nagy et al. [11]), or energy performance models (as in Gerber and Lin [9]); and secondly, by presenting a general technical view of how the generative design system should function instead of focusing on the overarching design process (as in Abrishami et al. [12]) or the specific prototype (as in Abrishami et al. [13]).

4.2. Additional Considerations

The findings from this study also shed light on multiple facets of generative design applications. The first of these is the importance of how a solution space is defined and how it should be given careful consideration not to be under- or over-constrained. An example of potentially wanting to over-constrain a solution space was seen during the project regarding the habitable area. As the case company entered the project, they had a set number in mind for the habitable area; however, during the project (and after receiving input from other studies performed on the same case) the case company realized that their initial target should be reconsidered, potentially in favor of a higher target. This highlights an example of where a priori preferences might hinder exploration and identification of novel solutions. This is partly explained by the changes of preferences and desires that might occur over time as one learns from experiences and supports the use of more progressive preference management [27]. Simultaneously, it was recognized by the authors in this study that the solution space might have been defined too broadly. Finding a direction in the exploration proved a challenge, as multiple variations of solutions seemed similarly interesting. This created multiple regions in the solution space that showed promising solutions, but exploring them all would be subject to time restrictions. A potential solution to this could have been to reduce the range or number of design variables, to add more constraints, or to include more metrics. There is an inherent balance to be found here, where a reduced number of design variables or an increased amount of constraints have the potential to simplify the exploration process at the cost of potentially not finding novel solutions. Similarly, the inclusion of additional metrics, such as those commonly used during

a regular design process, can result in a better evaluation of solutions that have the potential cost of an increased complexity both in terms of the technical solution (i.e., calculating each metric) as well as in the analysis made by the users of a generative design approach. The exact selections of design variables, constraints, and metrics are considerations that might differ from project to project. Thus, the framework was designed to be open to the addition of other setups than those used in this study in order to reflect the design context of other building projects. The application of generative design is also dependent on the targeted level of detail of the final solution set and how its potential to explore different levels of details. In line with this, Abrishami et al. [13] exemplified a top-down exploration starting with a building envelope and further adding details as the exploration progressed. Top-down exploration has also been shown in this study, where an empty block is progressively populated with one or several buildings through different levels of detail. This could be extended to allow multiple stages of a generative design approach, with each stage increasing in detail (a workflow that is similar to that suggested by Singh and Gu [8]).

Given these findings, it should be recognized that defining a solution space could entail an iterative process in itself, and choosing the appropriate generator algorithm requires equal attention (also suggested by Singh and Gu [8]). This has the potential to require a lot from the user of a generative design approach, and it should be noted that the developed framework for design exploration should not be viewed as necessarily being a 'perfect fit' with today's workflows in architectural design or the construction industry. As digital tools are used to augment the design process, the designer can encounter new interactions with the medium that are different from traditional design thinking [28]. This postulates that the design process and the role of the designers are subject to change. With the inclusion of the developed framework, the designer's role is not limited to only creating and evaluating a design, but rather opportunities are generated for the designer to become a tool builder [29] where they control and develop tools that generate designs [8]. This ties in with the argument that a designer benefits from the methods used, and that black box effects should be avoided to ensure that the decisions made are aided sufficiently [30]. Given these caveats, it should be recognized that considerations regarding knowledge, resources, and investments needs to be made with the implementation of a generative design approach to ensure that its benefits can be better understood.

It should also be noted that the application of generative design imposes requirements on the system architecture, due to its inherent preference towards having automated generation and evaluating solutions [9]. This challenge can be exacerbated, depending on the metrics that are requested and the models and tools needed to evaluate the metrics. Because this is a prerequisite of the effective use of generative design, these challenges need to be addressed (e.g., through BIM integration [13] or master models [19]). The integration of a BIM-based approach can also be applicable from a perspective of providing input data to a generative design approach, similar to that suggested by Salimzadeh et al. [14] or Hamidavi et al. [15]. Using BIM data as inputs could potentially be a powerful tool during later stages of design, once BIM models have already been created by other disciplines. Furthermore, a BIM model could also be the potential output from a generative design approach, as suggested by Abrishami et al. [12]. Both of these perspectives can facilitate the integration of a generative design process into an already BIM-based design process by mitigating some potentially resource-intensive translations that are otherwise required. This is a topic that should be further addressed for the future development of the proposed framework.

In general, the project participants (including the property developers) were positive about using the novel method of design exploration using the developed generative design framework. They indicated that there was potential in the approach, especially when targeting the exploration of solutions guided by novel requirements and targets from themselves, clients, and regulations. Because of the very early stage in the development of the residential block, the property developer had not fully begun to realize their ideas for the project. As such, a more in-depth evaluation of the implementation of a generative design approach was not conducted. Furthermore, this paper was also limited by only exploring one application of the developed framework, using the case of the

residential block development and a selected set of metrics for the evaluation of each generated solution. Regardless of these limitations, the findings from the development of the framework and the subsequent demonstration provide valuable contributions to the further development of the research field of generative design in architectural design, which is still in its early stages.

5. Conclusions

This paper presents a framework for design exploration using generative design in architectural design. The framework was demonstrated through a prototype developed for the early conceptualization of a residential block in northern Sweden, where residential building envelopes were generated and automatically evaluated for habitable areas, views, variations, disposable areas, and sunlight hours. Compared to related research in generative design for architectural design, this research contributes by including a more generic framework that shows the technical workflow of the generative design system. It also contributes by further exploring the effects on the solution space imposed by constraints and top-down exploration.

The field of generative design and its application in an architectural context shows promise and has the potential to be a part of a future designer's toolkit. Further research is needed, however, to increase the knowledge on how and when generative design is most applicable both when it comes to what types of problems it is suited to, as well as what types of methods are most suitable.

Author Contributions: Conceptualization, J.M. and M.S.; methodology, J.M.; software, J.M.; investigation, J.M.; writing—original draft preparation, J.M.; writing—review and editing, M.S.; visualization, J.M.; supervision, M.S.; project administration, M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Swedish Agency for Economic and Regional Growth within Kiruna Sustainability Center.

Acknowledgments: The authors would like to express their gratitude to the case company for their efforts in providing data and discussing the case from multiple interesting perspectives.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Diaz, H.; Alarcón, L.F.; Mourgues, C.; García, S. Multidisciplinary Design Optimization through process integration in the AEC industry: Strategies and challenges. *Autom. Constr.* **2017**, *73*, 102–119. [\[CrossRef\]](#)
2. Krish, S. A practical generative design method. *Comput. Des.* **2011**, *43*, 88–100. [\[CrossRef\]](#)
3. Evins, R.; Pointer, P.; Vaidyanathan, R.; Burgess, S. A case study exploring regulated energy use in domestic buildings using design-of-experiments and multi-objective optimisation. *Build. Environ.* **2012**, *54*, 126–136. [\[CrossRef\]](#)
4. Kalua, A. Envelope Thermal Design Optimization for Urban Residential Buildings in Malawi. *Buildings* **2016**, *6*, 13. [\[CrossRef\]](#)
5. Buyle, M.; Audenaert, A.; Braet, J.; Debacker, W. Towards a More Sustainable Building Stock: Optimizing a Flemish Dwelling Using a Life Cycle Approach. *Buildings* **2015**, *5*, 424–448. [\[CrossRef\]](#)
6. Jalilzadehazhari, E.; Vadiiee, A.; Johansson, P. Achieving a Trade-Off Construction Solution Using BIM, an Optimization Algorithm, and a Multi-Criteria Decision-Making Method. *Buildings* **2019**, *9*, 81. [\[CrossRef\]](#)
7. Shadram, F.; Mukkavaara, J. An integrated BIM-based framework for the optimization of the trade-off between embodied and operational energy. *Energy Build.* **2018**, *158*, 1189–1205. [\[CrossRef\]](#)
8. Singh, V.; Gu, N. Towards an integrated generative design framework. *Des. Stud.* **2012**, *33*, 185–207. [\[CrossRef\]](#)
9. Gerber, D.J.; Lin, S.H.E. Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. *Simulation* **2014**, *90*, 936–959. [\[CrossRef\]](#)
10. Shea, K.; Aish, R.; Gourtovaia, M. Towards integrated performance-driven generative design tools. *Autom. Const.* **2005**, *14*, 253–264. [\[CrossRef\]](#)
11. Nagy, D.; Lau, D.; Locke, J.; Stoddart, J.; Villaggi, L.; Wang, R.; Zhao, D.; Benjamin, D. Project discover: An application of generative design for architectural space planning. In Proceedings of the Symposium on Simulation for Architecture and Urban Design, Toronto, ON, Canada, 22–24 May 2017; pp. 1–8.

12. Abrishami, S.; Goulding, J.; Rahimian, F.P.; Ganah, A. Virtual generative BIM workspace for maximizing AEC conceptual design innovation: A paradigm of future opportunities. *Constr. Innov.* **2015**, *15*, 24–41. [\[CrossRef\]](#)
13. Abrishami, S.; Goulding, J.; Rahimian, F. Generative BIM workspace for AEC conceptual design automation: Prototype development. *Eng. Constr. Arch. Manag.* **2020**. [\[CrossRef\]](#)
14. Salimzadeh, N.; Vahdatikhaki, F.; Hammad, A. Parametric Modelling and Surface-specific Sensitivity Analysis of PV Module Layout on Building Skin Using BIM. *Energy Build.* **2020**, *216*, 109953. [\[CrossRef\]](#)
15. Hamidavi, T.; Abrishami, S.; Hosseini, M.R. Towards intelligent structural design of buildings: A BIM-based solution. *J. Build. Eng.* **2020**, *32*, 101685. [\[CrossRef\]](#)
16. Nagy, D.; Villaggi, L.; Benjamin, D. Generative urban design: Integrating financial and energy goals for automated neighborhood layout. In Proceedings of the Symposium on Simulation for Architecture and Urban Design, Delft, The Netherlands, 4–7 June 2018; pp. 265–274.
17. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [\[CrossRef\]](#)
18. Yin, R.K. *Case Study Research and Applications: Design and Methods*, 5th ed.; SAGE Publications: Thousand Oaks, CA, USA, 2014.
19. Sandberg, M.; Mikkavaara, J.; Shadram, F.; Olofsson, T. Multidisciplinary optimization of life-cycle energy and cost using a BIM-based master model. *Sustainability* **2019**, *11*, 286. [\[CrossRef\]](#)
20. Emami, N. Untangling parameters: A formalized framework for identifying overlapping design parameters between two disciplines for creating an interdisciplinary parametric model. *Adv. Eng. Inform.* **2019**, *42*, 100943. [\[CrossRef\]](#)
21. Shadram, F.; Mikkavaara, J. Exploring the effects of several energy efficiency measures on the embodied/operational energy trade-off: A case study of Swedish residential buildings. *Energy Build.* **2019**, *183*, 283–296. [\[CrossRef\]](#)
22. Kolarevic, B.; Malkawi, A.M. *Performative architecture: Beyond instrumentality*; Spon Press: New York, NY, USA, 2005.
23. Roudavski, S. Towards morphogenesis in architecture. *Int. J. Arch. Comput.* **2009**, *7*, 345–374. [\[CrossRef\]](#)
24. Chien, S.-F.; Fleming, U. Design space navigation in generative design systems. *Autom. Constr.* **2002**, *11*, 1–22. [\[CrossRef\]](#)
25. Koutamanis, A. Digital architectural visualization. *Autom. Constr.* **2000**, *9*, 347–360. [\[CrossRef\]](#)
26. Prats, M.; Lim, S.; Jowers, I.; Garner, S.W.; Chase, S. Transforming shape in design: Observations from studies of sketching. *Des. Stud.* **2009**, *30*, 503–520. [\[CrossRef\]](#)
27. Coello, C.A.C.; Lamont, G.B.; Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-objective Problems*; Springer: New York, NY, USA, 2007.
28. Oxman, R. Theory and design in the first digital age. *Des. Stud.* **2006**, *27*, 229–265. [\[CrossRef\]](#)
29. Aish, R. Extensible computational design tools for exploratory architecture. In *Architecture in the Digital Age*; Spon Press: New Yourk, NY, USA, 2003; pp. 344–353.
30. Brans, J.P.; Mareschal, B. The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decis. Support. Syst.* **1994**, *12*, 297–310.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).