Conceptual Structural System Layouts via Design Response Grammars and Evolutionary Algorithms

Sjonnie Boonstra[a,*], Koen van der Blom[b], Hèrm Hofmeyer[a,*], Michael T. M. Emmerich[b]

[a] *Eindhoven University of Technology, The Netherlands*
[b] *Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands*

Abstract

Two new methods to generate structural system layouts for conceptual building spatial designs are presented. The first method, the design response grammar, uses design rules—configurable by parameters—to develop a structural system layout step by step as a function of a building spatial design's geometry and preliminary assessments of the structural system under development. The second method, design via optimizer assignment, uses an evolutionary algorithm to assign structural components to a building spatial design's geometry. In this work, the methods are demonstrated for two objectives: minimal strain energy (commonly used objective for structural topology optimization) and minimal structural volume. In a first case study three building spatial designs have been subjected to the methods: Design via optimizer assignment yields a uniformly distributed Pareto front approximation, which incorporates the best performing layouts among both methods. On the other hand, results of the design response grammar show that layouts that correspond to specific positions on the Pareto front (e.g. layouts that perform well for strain energy), share the same parameter configurations among the three different building spatial designs. By generalizing, specific points on the Pareto front approximation have been expressed in terms of parameter configurations. A second case study addresses the use of a generic material and generic dimensions in the assessment of structural system layouts, which appears impractical. The study, therefore, applies a technique similar to topology optimization to optimize the material density distribution of each individual structural component, which can be regarded as a part of determining materials and dimensions in more advanced stages of the design of a system layout. This optimization approach is applied to the layouts that are part of the Pareto front approximations as found by the evolutionary algorithm in the first case study, the study shows that—after optimization—the fronts remain the same qualitatively, suggesting that the methods produce results that are also useful in more advanced design stages. A final case study tests the generalization that is established in the first case study by using the found configurations for the design response grammar, and it is shown that the generated layouts indeed are positioned near the desired positions on the Pareto front approximation found by the evolutionary algorithm. Although the evolutionary algorithm can find better performing solutions among a better distributed Pareto front approximation, the design response grammar uses only a fraction of the computational cost. As such it is concluded that the design response grammar is a promising support tool for the exploration and structural assessment of conceptual building spatial designs. Future research should focus on more types of structural elements; more objectives; new constraints to ensure feasible solutions, especially stress constraints; and the application of state-of-the-art techniques like machine learning to find more generalizations.

Keywords: Building Spatial Design, Multi-Disciplinary Design, Design Grammar, Structural Design, Automated Design, Design Optimization, Conceptual Design

1. Introduction

Building design has been an optimization task for centuries. During the early days in the field of building design, the primary struggle was to satisfy the basic objectives of a dry and warm shelter. Whereas nowadays, advances in experience and technology have made it possible to also include other objectives, e.g. aesthetics, comfort, material usage, and/or energy performance. As a consequence, the built environment has seen a sophisticated distribution into disciplines. Today, engineers can reach the limits of optimality within the scope of their discipline. However, trade-offs exist between disciplines. Therefore engineers need to accept concessions on the optimality of their design. Unfortunately, many engineers do not have an influence on some of the concessions that they must accept. This is because they are only involved in one of the later stages of the building design process, while the initial design stage contains the most critical design choices for many disciplines (Wang et al., 2002; Brown and Mueller, 2016). Even if engineers from all of the required disciplines would be included in the initial design stage, the challenging communication, the complex inter-disciplinary design relations, and the complex trade-offs between disciplines would still complicate the optimization process in building de-

*Corresponding author

Email addresses: s.boonstra@tue.nl (Sjonnie Boonstra), k.van.der.blom@liacs.leidenuniv.nl (Koen van der Blom), h.hofmeyer@tue.nl (Hèrm Hofmeyer), m.t.m.emmerich@liacs.leidenuniv.nl (Michael T. M. Emmerich)

sign Haymaker et al. (2004).

Structural design is one of the two disciplines—together with architecture (aesthetics/spatial design)—that shapes building designs the most during their early design stages. These two disciplines also interact strongly: a spatial design can only exist or be experienced when it is realized by a structure. On the other hand, the structure inevitably influences the spatial design, because it occupies some of its space and it also affects the aesthetics of the building design (Khemlani et al., 1998). Nonetheless, in practice, the design process starts by only considering the building spatial design, because many of the functional requirements in a design brief are defined by the spatial design alone. Including structural design at the beginning of the design process can, however, lead to savings in material use and lead to structural design solutions that are aesthetically pleasing. To this aim, methods can be developed that provide a conceptual spatial design with a structural system layout, and this system layout can be assessed. As such, the suitability of a conceptual spatial design can be determined with respect to its structural potential, and the system layout itself may be a candidate for further design developments. In this paper two such methods are presented and compared: (I) a design response grammar, which uses design rules—configurable by parameters—to develop a structural system layout step by step as a function of a building spatial design's geometry and preliminary assessments of the structural system under development. (II) design via optimizer assignment, which uses an evolutionary algorithm to assign structural elements to a building spatial design's geometry. Both methods generate structural system layouts for conceptual building spatial designs, and inevitably need objectives to assess these layouts. For demonstration purposes here minimal strain energy (commonly used for structural topology optimization) and minimal structural volume are used. Using an evolutionary algorithm, design via optimizer assignment yields a Pareto front approximation, which contains information regarding trade-offs between the objectives. Via a parameter study, it is demonstrated that the parameters of the design response grammar can be configured such that desirable positions on the Pareto front (e.g. a layout that performs well for strain energy) can be found. Using these configurations, specific sets of solutions can be generated quickly, which is not possible with an evolutionary algorithm. The presented methods can be used to provide architects insight into the locations within a conceptual building spatial design where placement of structural elements is logical or expected. Moreover, they can serve comparative assessment—from a structural engineering point of view—of conceptual building spatial designs, without the need to define or assume detailed design information. Additionally, they can support structural engineers in their task to design, optimize, and decide on structural system layouts for complex building spatial designs. Finally, the methods support multi-disciplinary building optimization, in which many conceptual spatial designs have to be evaluated—within a limited amount of time—for their potential in structural performance (Boonstra et al., 2018).

This paper continues with an overview of the background, the related work, and a motivation for the presented work, in section 2. Following that, section 3 presents the methodology that is used for the two new methods. Then in section 4, the two new methods are studied in three cases studies. After a discussion in section 5, in section 6 the conclusion and outlook of the presented work are given.

2. Background and Related Work

This section starts with discussing optimization in general. Next, it elaborates on multi-disciplinary design (optimization) in the built environment. Subsequently, literature on early-stage building design support methods, and early-stage building design optimization are discussed. Finally, the motivation for this work is presented.

2.1. Optimization

A generic mathematical formulation for optimization problems is given in equation 1, in which there are ℓ objective functions $f_i(x)$. Here a possible solution is represented by $x \in X$, and X is the collection of all possible solutions, the so-called search space. A possible solution x is only considered if both all m inequality constraint functions $g_j(x)$, and all n equality constraint functions $h_k(x)$ hold.

$$\begin{aligned} \min_x : \quad & f_i(x), && i = 1, 2, \ldots, \ell \\ \text{subject to :} \quad & g_j(x) \geq 0, && j = 0, 1, \ldots, m \\ & h_k(x) = 0, && k = 0, 1, \ldots, n \end{aligned} \tag{1}$$

In the case of multiple objectives ($\ell > 1$), there is not a single optimal solution. In fact, with multi-objective optimization, a trade-off between objective functions often occurs. The best solution in one objective may not be good in the other objective(s). Optimality for multiple objectives is therefore formulated in terms of non-dominance. A solution x is dominated by solution x^* if both conditions in equation 2 are satisfied. Non-dominated solutions are those solutions that are not dominated by any other solution. When a solution cannot be improved in any one objective, without getting worse in another objective, it is a Pareto optimal solution. The set of all such solutions is called the Pareto front. Note that if a solution is non-dominated with respect to a subset of X, it is not necessarily part of the Pareto front. If only a subset of X is evaluated then the known set of non-dominated solutions is called the Pareto front approximation (PFA). For a more in-depth introduction to multi-objective optimization, and an overview of recent developments, the reader is referred to the work of Emmerich and Deutz (2018).

$$\begin{aligned} \forall i : f_i(x^*) \leq f_i(x) \\ \exists i : f_i(x^*) < f_i(x) \end{aligned} \tag{2}$$

2.2. Multi-Disciplinary Building Design

Trade-offs between disciplines in the built environment have been researched for several decades, an early example is the work by Gero et al. (1983). With increasing demands for optimality in building design, nowadays, multi-disciplinary research is receiving more and more attention. Some research on multi-disciplinary design (optimization) focuses on obtaining performance measurements during the design process, such that a designer can make informed design decisions. This is carried out by Welle et al. (2011) for example, who present a method to assess Building Information Models (BIM) on their performance. It can, however, be questioned how well designers can foresee the impact of their design decisions. In recent years, research has shifted towards providing designers with insights into the impact of the used design parameters. For example, Schlueter and Geyer (2018) aim to give designers feedback on the effect of and the relations between design parameters. Or, Hopfe et al. (2012) present a multiobjective optimization method to assess the impact of design parameters using Evolutionary Algorithms (EAs) and statistical sensitivity analysis. Moreover, Geyer and Schlueter (2014) introduce a method to create surrogate models from a BIM model to efficiently explore design parameters.

Research on multi-disciplinary building design is focused on more than just parameter impact. Another aim within the field is to make optimization methods more accessible to designers. This is the case in the work of Geyer (2009), where the quick exploration of the search space for a few possible structural models for a building spatial design gives an early insight into the structural performance. Other research suggests the use of specialized equation-based models for the evaluation of building performance (Wetter et al., 2016). Such models enable fast gradient-based optimization which makes them useful for real-time design support. Also, the choice and the correct application of optimization algorithms are of influence, as is illustrated by Hamdy et al. (2016). Tools that can enable designers to create viable designs for disciplines outside of their domains are developed as well. For example, a tool for the creation of a structural design within an architectural design environment is presented by Steiner et al. (2017). Or by Beghini et al. (2014), who integrated different design domains by applying a structure optimization algorithm (topology optimization) in their architectural design process.

Finally, other research on multi-disciplinary design investigates changes in one discipline that affect the search space of another discipline. This phenomenon is called co-evolution, examples of such behavior and suggested methods to research these are presented by Maher and Tang (2003). Research that takes into account co-evolution is not widespread, however, examples can be found, e.g. Hofmeyer and Davila Delgado (2015) consider it for building spatial design versus building structural design. They show that a simulation of a human co-evolutionary design process of structural design and building spatial design can quickly find solutions that are better than those found by an optimization algorithm. This is because their method can handle search spaces of arbitrary size, while that of an optimizer must be fixed and is often limited to keep computational times acceptable.

2.3. *Early Stage Building Design*

The performance of a design can be influenced most during the conceptual design stage. This statement is widely supported, for example, Wang et al. (2002) stress that the influence on the performance of a design is large at the beginning of a design process, but decreases rapidly as the design progresses. On the other hand, they conclude that the available number of design tools at the beginning of a design process is small, and only increases slowly. Design engineers therefore tend to only optimize their designs during the later stages, a statement which is also supported by Machairas et al. (2014). Although these optimization approaches still benefit a building's performance, there is a growing desire for optimization at the conceptual design stage (Okudan and Tauhid, 2008; Clevenger and Haymaker, 2012; Negendahl and Nielsen, 2015; Nielsen et al., 2016; Touloupaki and Theodosiou, 2017). Therefore, Clevenger and Haymaker (2011) and Basbagill et al. (2014) focus on ways to give designers feedback on the effectiveness of the parameters and the design methods that they use. However, a more fundamental approach is suggested by Chong et al. (2009), who describe the optimization of a conceptualized design. In their view, a designer should focus on how to conceptualize designs and design relations, instead of estimating the performance of sketch designs for decision support.

The literature is not limited to stressing the importance of early-stage design optimization, modeling support for conceptual building spatial designs is researched as well. For example, shape grammars are presented by Stiny (2006) and Ruiz-Montiel et al. (2014), to aid in modeling the building spatial design in the conceptual design stage. Algorithms to find the optimized layouts of a building spatial design have also been introduced (Liggett, 2000; Sharafi et al., 2017; Song et al., 2016).

Common methods for early-stage design support are based on performance computation. To give a number of examples, designs are parameterized and optimized for simple objective functions by Gerber and Lin (2014). Similarly, a simplified evaluation model for conceptual designs is presented by Picco et al. (2014). Ritter et al. (2015) simulate the building physics of conceptual design models via a plugin in a CAD environment, which provides users with design performances and parameter impacts.

Another common method is the use of tools that generate (a part of) a design during an early stage of the design process. This is particularly common for structural design. An explanation for this could be the fact that there is a high dependency between architectural and structural design disciplines. Examples for structural design support during the architectural design phase have been found (Rafiq and MacLeod, 1988; Fenves et al., 2000; Mora et al., 2008).

Research on early-stage building design does not focus solely on the performance of a conceptual design. For example, the work of Azzouz et al. (2017) applies life cycle analysis in a real-world case study to show the effects of early-stage design optimization on a real world building. Moreover, the available methods for early-stage design, as well as methods to monitor them during their lifetime, are reviewed by Oh et al. (2017). They try to make these methods more accessible for policy-makers and engineers. Embodied and operational energy are considered in an extensive study for long-span structures by Brown and Mueller (2016). Finally, Hopfe and Hensen (2011) discuss uncertainties in the performance of a building design regarding the determination of the impact of parameters.

2.4. Motivation

The available support for multi-disciplinary building design optimization in the conceptual design stage is still limited, whilst critical design decisions are made at this stage. This has sparked research for conceptual building design optimization, to which this paper contributes. Within the wider scope of this research, a toolbox was developed, which enables performance assessment of conceptual building spatial designs for their structural and building physics performances, as well as the modification of such designs (Boonstra et al., 2018). Some of the other work within the same framework includes a study on a co-evolutionary approach for optimized building spatial and building structural design in (Hofmeyer and Davila Delgado, 2015). And, in (van der Blom et al., 2016a,b, 2017) evolutionary algorithms are applied and configured to building spatial design optimization. Furthermore, in (Boonstra et al., 2017), a combination of co-evolutionary design simulations and evolutionary algorithms is proposed to be able to effectively explore and find optimal designs in large search spaces. In the aforementioned research, structural performance evaluations of building spatial designs were obtained from structural designs that were generated by algorithms that operate on simple design rules, termed structural design grammars. Such design grammars may place structure in places where it is not logical nor expected, and thus they may not account for the full potential of the structural performance of a given building spatial design. A logical placement of components within a structural system layout can be formulated as a material optimization problem, i.e. material should only be placed at locations where it is useful. The work in this paper aims to develop methods that, for a given conceptual building spatial design, can generate structural system layouts that perform well structurally seen. To assess performance, inevitably objectives are required, for which in this paper—for demonstration purposes—minimal strain energy and minimal structural volume are used.

When looking at the state-of-the-art, there appears to be a scarcity of methods that can generate structural designs for conceptual building spatial designs. Additionally, such methods usually require some form of interaction from a designer to solve problems with the generated design. This is not convenient when many possible solutions have to be assessed, e.g. when exploring a large search space, quick evaluations without human interaction are desirable. Additionally, a detailed model for an extensive structural analysis is not necessary when only a quick insight into the structurally relevant locations within a conceptual building spatial is desired. A structural design grammar is fast, but it typically does not place structure logically from a structural engineering point of view. The work in this paper therefore also aims to develop a method that can quickly generate structural system layouts that perform well for certain objectives. It should be stressed that, in this work, a solution entails a structural system *layout* with generic element dimensions and material properties, and so does not include the final dimensions and materialization. A solution is thus not a final stage structural design, and it is not intended to be, but instead it can offer insight in a structural concept that is required to realize a conceptual building spatial design. Nevertheless, the work in this paper also investigates if the proposed layouts are useful in a more advanced stage of the design process.

3. Methodology

This section discusses the methodology that is used for the presented work. First, the relevant aspects of an existing toolbox for building spatial design optimization are introduced. Thereafter, an existing structural design grammar that is directed by user input is introduced and elaborated. The existing grammar contains details that will support the introduction of two new methods to generate a structural system layout in the last two subsections of this section. The first method, the design response grammar, can be calibrated by parameters and uses design rules that operate on the geometry of a building spatial design and an analysis of a preliminary structural model to develop a structural system layout. The second method, called design via optimizer assignment, uses an evolutionary algorithm to assign structure to the geometry of a building spatial design.

3.1. Toolbox

As discussed in the motivation (section 2.4), the presented work is part of a broader research scope that focuses on building spatial design optimization. In this context, a multipurpose toolbox to support building spatial design optimization has been developed (Boonstra et al., 2018). The toolbox functions that are relevant to the scope of this paper are briefly discussed in the following.

3.1.1. Spatial Design

Optimization requires a formal representation of the design problem, and in the toolbox, building spatial designs are therefore formalized as follows. A building spatial design is defined by one or more spaces that are each specified with six variables (not considering metadata like an ID or other characteristics). Specifically, these are: The location of a space (x-, y-, z-coordinates of the base); And, a space's dimensions (width, depth, and height). A building spatial design in the toolbox is therefore limited to cuboid spaces in an orthogonal grid. This orthogonality is applied for the sake of clarity and simplicity, however, the methods that are researched using the toolbox need not adhere to this limitation in the later stages of their development. Additionally, $z = 0$ is set to represent the ground surface and values below zero ($z < 0$) are underground.

In the toolbox, two levels of building spatial design information are identified: the geometry level and the building design level. On the geometry level, a design is decomposed into the following geometry entities: cuboids, rectangles, line segments, and vertices. This decomposition is performed such that no intersections exist between any geometry entities. On the building design level, a spatial design is decomposed into the following building design entities: spaces, surfaces of-, edges of-, and points of spaces. Such a distinction between geometry and design is useful when a discipline-specific design needs to be defined, for example, structural design components such as flat shells are defined using geometry entities. This is to make sure that in a structural model all nodes of adjoining structural components are coincident. However, the live loading on the structural model is defined using building design entities. This is because live loading is defined per space. The two levels of design and the given examples have been illustrated in figure 1.

In the toolbox, the two levels of design come together in the so-called building conformal model; figure 2 depicts the UML class diagram of this model. The building conformal model links all the different entities in each level of design with each other. For example, a surface is realized by four edges and—together with five other surfaces—it realizes a space. At the same time, a surface can be associated with one or more rectangles, whereas a rectangle can belong to one or two surfaces, etc. This is useful, for example, when structural design components that are defined by geometry entities have to be loaded with loads that are defined by building design entities. For more information the reader is referred to (Boonstra et al., 2018).

3.1.2. Structural Analysis

Structural analysis is implemented in the toolbox to be able to evaluate the structural models that are created. The analysis is performed using the finite element method, for details on the implementation the reader is referred to (Boonstra et al., 2018). A structural model in the toolbox can consist out of the following structural components: flat shells, beams, trusses, loads, and constraints. Before analysis, each component is meshed (divided) into n^d elements, where n, the mesh size, is the number of elements in each dimension and d is the dimensional size of a component (e.g. a column is 1-dimensional and a flat shell 2-dimensional). Finally, a numerical analysis (termed finite element analysis) computes the deformations of the structure, which—together with the structural system—can be used to calculate other design responses.

Structural design is a complex process and it is possible that a design grammar generates a structurally unstable solution. Structural models can, therefore, be subjected to a stability check, which is performed as follows. First, to save computation time, the model is meshed without its loads but with its constraints using a mesh size $n = 1$. Accordingly, it is checked whether the solver (the Simplicial-LLT solver of the Eigen C++ library; Guennebaud et al., 2019) can successfully decompose the global stiffness matrix of the finite element model. Here, the stiffness matrix is the numerical system that represents the structural model (for more details on the stiffness matrix see Boonstra et al., 2018). If the stiffness matrix of a model cannot be decomposed, it is considered unstable, the performance of such structural models can then be penalized, or even be disregarded altogether.

3.1.3. Clustering

Clustering can help select building spatial designs or parts of a building spatial design based on similarities. For example, in the toolbox, a building spatial design can be modified based on its performance: spaces with poor performance are removed and spaces with good performance are split into multiple new spaces. In such cases, it is desirable that spaces with similar performance are selected together for modification. This prevents arbitrary phenomena like numerical errors or the order in computer memory to play a role in the selection. Moreover, using clustering, possible symmetries in a building spatial design are preserved during the modification. K-means clustering, as found in e.g. (MacKay, 2003), has been implemented in the toolbox. Clustering parameters that need to be specified are: The bounds for the cluster size k_{min} and k_{max}; And, the number of runs l per cluster size. This results in $(k_{max} - k_{min} + 1) \times l$ possible divisions in clusters, out of which only one is selected as follows. The quality of a clustering is defined by the sum of the variance within each cluster $\sigma_{sum,k} = \sum_{i=1}^{k} \sigma_i$, where a lower value indicates a clustering of higher quality. For each cluster size k over all runs l, the clustering that has the lowest value for $\sigma_{sum,k}$ is stored. Accordingly, the second order change of $\sigma_{sum,k}$ is computed for each cluster size k, according to equation 3. Note that, in order to calculate this value for k_{min} and k_{max}, two additional cluster sizes must be computed: $k_{min} - 1$ and $k_{max} + 1$. The clustering size (k) with the largest value for $\sigma''_{sum,k}$ is then selected as the best performing clustering size.

$$\begin{aligned} \sigma''_{sum,k} &= (\sigma_{sum,k+1} - \sigma_{sum,k}) - (\sigma_{sum,k} - \sigma_{sum,k-1}) \\ &= \sigma_{sum,k+1} + \sigma_{sum,k-1} - 2\sigma_{sum,k} \end{aligned} \quad (3)$$

3.2. Design Grammar Directed by User Input

Here a design grammar is defined as a set of design rules that operates on the building conformal model of a building spatial design in order to generate a discipline-specific design. The grammar that is presented in this section can create structural design models based on user input. First the procedure that the grammar follows is outlined, then the processing of user input is discussed, and finally, an explanation of how a structural model is generated is given. Note that the two new methods to generate a structural model (to be introduced after this) use many of the concepts explained in this section.

3.2.1. General Procedure

As presented in the section on structural analysis in the toolbox, a structural model consists of a combination of the following structural components: flat shells, beams, trusses, loads, and constraints. To generate these, two types of so-called rule sets are defined for the grammar. One rule set that operates on the rectangles, and one that operates on the line segments of a building conformal model. Note that both rule types operate on geometry entities (figure 2). The rule sets first check which type of structural component (flat shell, beam-, truss-, or no component) should be generated. To that end, for each type of structural component, the rules check the information that is contained within the geometry entities and building design entities against the information that is given in user-defined input files. When a check is positive, a component is added to the structural model, otherwise, nothing is added. After initializing a structural component it is checked whether or not loads and/or constraints should be applied to that component.

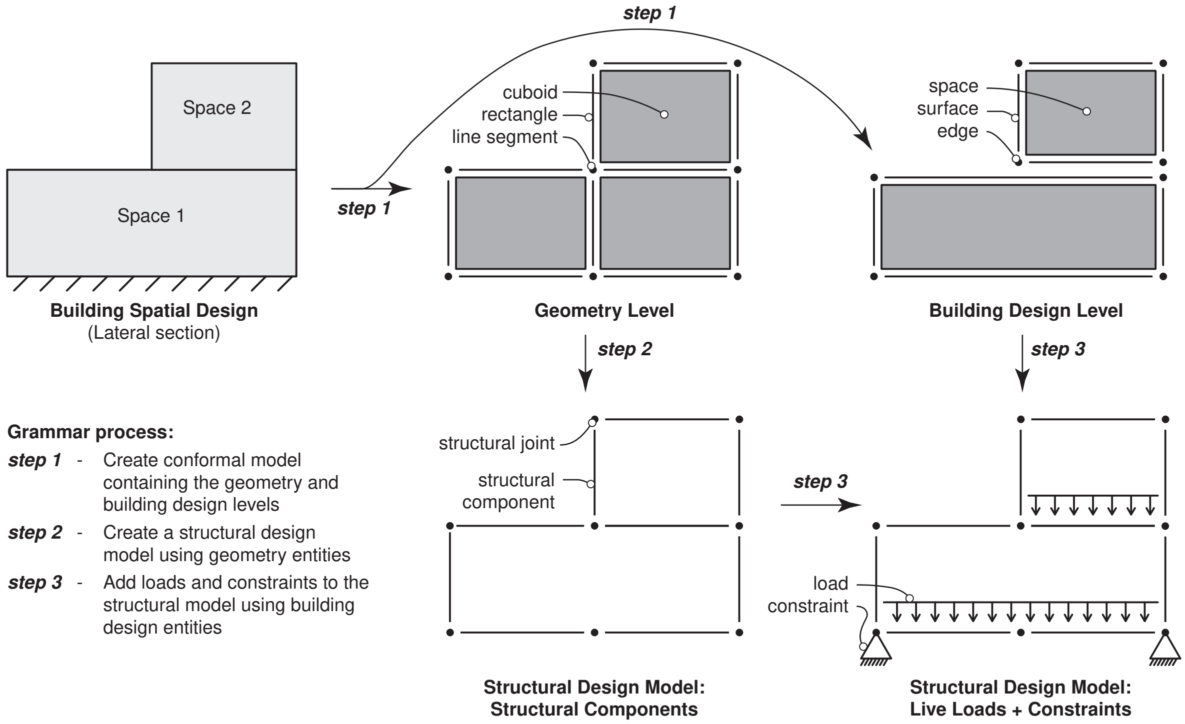

Fig. 1. The procedure through which a grammar assigns structural components to the geometric and building design entities of a building spatial design.

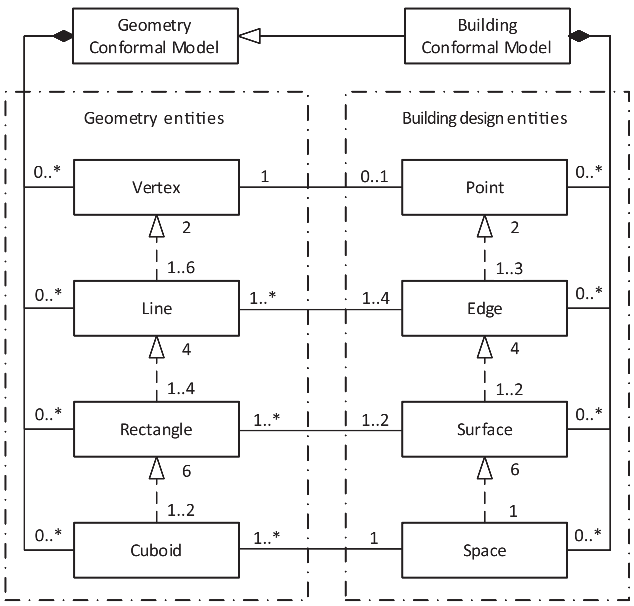

Fig. 2. UML class diagram of the building conformal model.

3.2.2. *User Input*

A user of the toolbox can describe the structural design that is created by the design grammar by specifying several options in input files. First of all, a structural design settings file is required, in which the structural loads, components (e.g. flat shells, beams or trusses), and their properties are defined. Users can as such define all the building blocks for the structural model that they intend to use for their structural design. Secondly, users can assign structural types to spaces and/or surfaces in the building spatial design to specify what structure is placed at the corresponding locations. However, at some locations, two different types of structure may be assigned. Therefore, in a third input file, users can specify the choice of structure when conflicting structures would be placed at the same location.

3.2.3. *Creation of a Structural Model*

The core purpose of the grammar is the creation of a structural model, how this is carried out is explained next. The grammar starts by checking for each rectangle what type of structure should be generated at its location. To that end, four structural types can be assigned to a rectangle: a `flat shell`, `beam`, `truss`, or `no structure`. Figure 3 depicts the structural components that are generated for each structural type assignment. Note that adjoined rectangles with different structural type assignments can create a conflict in the adjoined region. The generation of structural components is therefore split into two, first, a rectangle's area is evaluated with the so-called *"rectangle rules"* and accordingly the rectangle's line segments are evaluated with the so-called *"line segment rules"*.

Rectangle rules. The design grammar starts by applying a rectangle rule set for each eligible rectangle before handling any line segment rules. A rectangle is eligible for a rectangle rule set if it is associated to one or two surfaces (within the context of a building conformal model, figure 2). If it is eligible, then the rectangle rule set will first classify the rectangle into a floor or a wall. This classification is carried out by checking if the absolute value of the angle between the rectangle's normal vector $\mathbf{n}$ and the unit vector $\hat{\mathbf{k}} = [0\ 0\ 1]^\top$ is larger than 45°

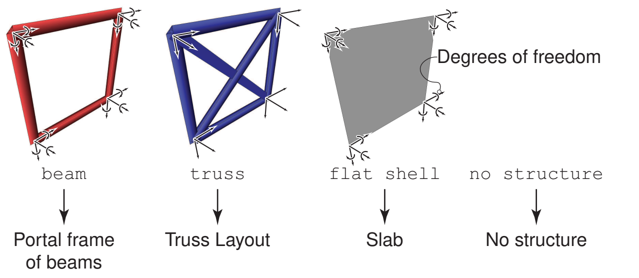

Fig. 3. The different structural types in the toolbox that can be assigned to a rectangle. Note that boundary conditions are applied to the structure in a later stage of the grammar.

degrees ($\left|\mathbf{n} \angle \hat{\mathbf{k}}\right| > 45°$). If this holds, then the rectangle is classified as a floor, otherwise, it is classified as a wall. From the user input, it is then determined which structural type applies to the rectangle. If the structural type is `flat shell`, then a flat shell is initialized. If it is `truss`, then two diagonal trusses are initialized. Finally, if any other type is selected, nothing is initialized.

After generating a structural component for a rectangle it is checked whether or not a surface load should be applied. In a structural design settings file, a user can specify a load case, a direction, and a type for each defined load. The possible load types are: wind pressure, wind shear, wind suction, and live load (floor load). For a rectangle, wind loading is only considered if it is associated to exactly one surface and if the maximum z-coordinate of that rectangle is larger than zero. Or in other words, when it has exactly one adjacent space and is located above the ground surface ($z \geq 0$). Wind loading is then applied according to table 1 and equation 4. Here α_r is the angle (in the half open interval: $[0°, 360°)$) between the unit vector $\hat{\mathbf{j}}$ ($[0\ 1\ 0]^\top$) and the xy-plane projection of the rectangle's outward facing normal, α_w is the angle between $\hat{\mathbf{j}}$ and the wind direction vector (which is only defined in the x- and y-directions). Live loading is applied whenever a rectangle is specified as a floor, note that this will also lead to a live load on the roof of a building.

$$\beta = \begin{cases} |\alpha_r - \alpha_w|, & if\ |\alpha_r - \alpha_w| \leq 180° \\ 360° - |\alpha_r - \alpha_w|, & otherwise \end{cases} \tag{4}$$

Table 1. Table with conditions for wind load application.

wind load type	condition
pressure	$90° < \beta^1 \leq 180°$
suction	$0° \leq \beta^1 < 90°$
shear	$90° \leq \beta^1 \leq 180°$ or rectangle is floor

[1] β is given by equation 4

When a surface load is assigned to a rectangle, it is possible that no structure exists in the structural model to which that load can be applied. A low stiffness flat shell component will then be placed in the structural model at the rectangle's location. A low stiffness will prevent an influence on the overall stiffness of the structural model, while it can still appropriately transfer the loads to the bearing components in the model. This is analogous to a real-world scenario where there is no structure behind a façade and wind loads are transferred to the structure via that façade, without the façade taking part in the building's structural system. A convergence study has shown that a factor of 1e−6 is a sufficient reduction in the order of the smallest elasticity modulus that is used within the structural design model, without affecting the structure's stiffness nor introducing numerical discrepancies to the model. The low stiffness components are ignored in the final stages of the structural analysis, i.e. when visualizing the structural design and when computing the performance of a design.

Line segment rules. A line segment rule set is only applied to those line segments that are associated with at least one rectangle for which a rectangle ruleset was created. The rule set for a line segment starts by iterating through each of its associated rectangles, rectangles for which no rule set was created are skipped. Each iterated rectangle is then checked for its structural type, i.e. `flat shell`, `beam`, `truss`, or `no structure`. This type is also assigned to the considered line segment. However, a ranking is applied in case of conflicting types between the iterated rectangles: `flat shell` over `beam`, `beam` over `truss`, and `truss` over `no structure`. Whenever a line segment is assigned the structural type of a rectangle, the properties that are associated with that rectangle and structural type are also applied. A structural component is generated in the structural model at the location of the line segment accordingly. When the conversion type is `beam` or `truss`, then respectively a beam or truss is initialized. For other types, nothing is initialized. Figure 4 gives a demonstration (2D) of the generated structural model after the assignment of structural types to a building spatial design. The figure also illustrates the ranking that is applied in case of conflicting structural types in adjoining regions, e.g. no truss or beam components are present at the border of a flat shell.

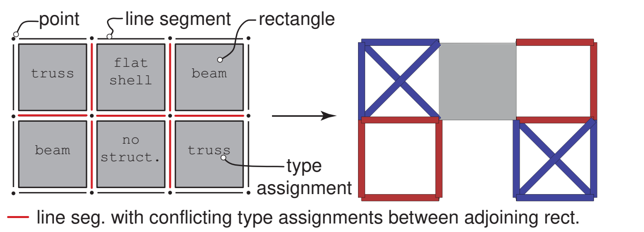

Fig. 4. Generated structure based on structural type assignments of rectangles, also note type assignment at adjoining line segments in-between rectangles.

Constraints are applied in the last step of the line segment rules. If a line segment belongs to a rectangle that has been classified as a floor and the z-coordinates of its vertices are less than or equal to zero, a line constraint is applied for each displacement degree of freedom (movement in x-, y-, and z-direction). If the structural type specifies no structure for the line segment, only the structure coinciding at vertices are constrained at the location of these vertices.

3.3. Design Response Grammar

A grammar is convenient when many building spatial designs need to be assessed for their structural response, which would be a slow process if user interaction is required. Simple design rules can be used to obtain a structural system layout, but such rule sets may place structure at locations where it is not logical nor expected. Here the design response grammar is proposed as a method that can quickly develop layouts in which the structure is placed in sensible locations. This grammar also uses design rules, but instead of solely operating on the geometry of a building it also operates on a design response, which is computed from a preliminary model of the layout that is under development. The design response grammar can be configured by parameters that allow control over the different types of design response and structure that will be considered by the grammar. Additionally, the design response grammar can be configured such that a layout is developed in a step by step manner, in which at each step structure is generated based on a design response obtained from the unfinished model that has been created by the preceding steps so far. In doing so, complex design rules are avoided, which would be necessary if structurally sensible solutions would be generated solely based on geometry. In this section, first the used design response grammar is introduced, and then the algorithm and its parameters through which the grammar creates a structural model are explained.

3.3.1. Design Response

A building spatial design by itself does not have a structural response. Therefore, a preliminary structural design model—termed substitute model—is introduced, which can be analyzed to yield a design response. The substitute model is created by placing a so-called substitute component at the location of each rectangle that is associated with a surface in the building conformal model (see figure 2 for associations in the building conformal model). In the grammar, substitute components will be replaced by beams, trusses, flat shells, or nothing, this replacement is based on their design response. The design response that is used here is the strain energy of a substitute component. To be able to use the substitute component in the existing design grammar structure of the toolbox, a new structural type is defined: `substitute`. The rectangle and line segment rules apply to `substitute` in the same way as other types, it ranks last with the type assignment in the line segment rules. A substitute component is similar to the low stiffness flat shell components that are used for the application of surface loads in the rectangle rule sets (section 3.2.3). Also here, a low stiffness enables the uncompleted structure to be analyzed without affecting its structural behavior.

Separated strain energies. The four different structural types (figure 3) can be used to replace a substitute component. From an engineering point of view, each structural type is well-suited for a certain type of loading, e.g. a truss layout is suitable for shear loading, a portal frame of beams is suitable for in-plane normal loading, and a flat shell is (among others) suitable for out-of-plane loading. To identify which type of loading is predominant within a substitute component, its stiffness term is separated into three terms: bending, normal, and shear. In the toolbox, the out-of-plane behavior (bending) of the flat shell element formulation is already derived separately. However, to obtain the formulation for the two separate types of in-plane behaviour, the constitutive relation is split in two terms according to equation 5 (where ν is the Poisson ratio and E the elasticity modulus [N mm^{-2}]). Using these separated formulations, the strain energies of the elements are computed for each type of loading: U_{sep} ($sep \in \{shear, norm, bend\}$). For more information on the used element formulations and derivations of these formulations, the reader is referred to (Boonstra et al., 2018).

$$
\begin{aligned}
&\overbrace{\frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}}^{\text{total}} = \\
&\overbrace{\frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & 0\end{bmatrix}}^{\text{normal}} + \overbrace{\frac{E}{1-\nu^2}\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}}^{\text{shear}}
\end{aligned}
\tag{5}
$$

3.3.2. Creation of a Structural Model

The design response grammar, see algorithm 1, uses an iterative process to generate the structural model. It starts by assigning the `substitute` type to every rectangle that is associated with one or two surfaces. Then, each iteration starts with the generation of a structural design model using the rectangle and line rules (section 3.2.3). After initialization, the structural design model is meshed and analyzed. Every i^{th} iteration, each `substitute` rectangle j—i.e. each rectangle that is assigned the `substitute` type—is subsequently clustered by its total design response, which is the total strain energy $U_{tot,i,j} = \sum U_{sep,i,j}$ obtained from the structural analysis. A criterion to limit the number of iterations is introduced in equation 6. Here $\eta_{conv} \in \mathbb{N}$ denotes the maximum number of iterations, $n_{subs,0}$ the initial amount of `substitute` rectangles, and $n_{subs,i}$ the number of `substitute` rectangles at the i^{th} iteration. If this criterion is not satisfied then the rectangles in the cluster with the highest mean compliance will be substituted (as described in the next paragraph) and the cluster is then removed. This is repeated until the convergence criterion is satisfied, the iteration is then ended. The iterative process is repeated until there are no more `substitute` rectangles left in the structural design model. Note that the substitution of rectangles of a large cluster may result in $n_{subs,i}$ being so small that the criterion in equation 6 is already satisfied before the next iteration, in that case—in the implementation—the next iteration $(i + 1)$ is skipped. Moreover, in the final iteration, clustering of the `substitute` rectangles is superfluous and it is therefore—in the implementation—skipped.

$$
n_{subs,0} - \left\lceil \frac{n_{subs,0}}{\eta_{conv}} \right\rceil \cdot i < n_{subs,i} \tag{6}
$$

Substitution. When a `substitute` rectangle is selected to be replaced by a new structural type, first all strain energies ($U_{tot,i,j}$ and $U_{sep,i,j}$) are computed. Following that, the strain energy of the `substitute` rectangle $U_{tot,i,j}$ is compared to a fraction η_{noise} of the mean strain energy in the initial structural model $U_{tot,mean,0}$, which can be found according to equation 7. This check is introduced to avoid type assignments based on numerical noise when the magnitude of the design response is small. If it is lower, the rectangle is assigned the `no structure` type. Otherwise, a new type will be assigned based on equation 8, which consists of the ratio U_{sep}/U_{tot}, and a predefined threshold $\eta_{sep} \in [0.0, 1.0] \in \mathbb{R}$. If equation 8 holds for bending strain energy, the rectangle is assigned `flat shell`; if it holds for normal strain energy it is assigned `beam`; if it holds for shear strain energy it is assigned `truss`. Note that the order of these checks is important because the equation might hold for multiple types of strain energy, but only one type of structural element can be assigned. This is why each check is performed in a predefined order, and as soon as one of them holds the others that follow will no longer be evaluated. When none of the three hold, the default type `no structure` is assigned to the rectangle. The checking order is stored in the set $\mathbf{c}$ which can be any permutation of $\{1, 2, 3\}$, where 1 activates the check on shear strain energy, 2 the check on bending strain energy, and finally 3 the check on normal strain energy.

$$U_{tot,mean,0} = \frac{\sum_{i=0}^{n_{subs,0}} U_{tot,0,i}}{n_{subs,0}} \tag{7}$$

$$\frac{U_{sep}}{U_{tot}} \geq \eta_{sep} \tag{8}$$

The process of the design response grammar is illustrated in figure 5 for an arbitrary building spatial design. In this example, a structure is created in two iterations for a building with three spaces. For illustrative purposes, the remaining parameters of the grammar have been selected for this example such that each structural type is assigned in the final design at least once.

3.4. Structural Design via Optimizer Assignment

This section presents an assignment function in the toolbox that an optimizer can use to assign structural types in its search for optimal structural system layouts for a given building spatial design. An optimizer is applicable because a structural design in the toolbox is created using a building conformal model, which has a fixed number of entities that can be assigned a structural type (`beam`, `truss`, `flat shell`, or `no structure`). If ordered in a string, the assigned types form a set of parameters similar to genomes in the field of evolutionary optimization. First the assignment and the genome are discussed, thereafter, a suitable optimizer is proposed, and finally, the objectives and constraints are discussed.

3.4.1. Assignment Function and Genome

The assignment function operates as a black-box objective function for the optimizer by taking a string of design variables as input and returning the objective values as output. Each input variable represents the choice of the structural type for one rectangle. As such, the genome should contain the same number of variables as the number of rectangles that are associated to one or two surfaces in the building conformal model. The set of valid variable values is $\{1, 2, 3, 4\}$. Here "1" assigns `no structure` to a rectangle, "2" assigns `truss`, "3" assigns `beam`, and "4" assigns `flat shell`. The order in which the genome assigns types to rectangles is determined by the order in which the eligible rectangles are stored in the building conformal model. After the assignment, the rectangle and line segment rules are applied to generate the structural design model (section 3.2.3), which is then evaluated to obtain the objective values. Finally,

Algorithm 1 Iterative replacement of substitute components

for each rectangle belonging to one or two surfaces **do**
    Assign `substitute`
end for
$i = 0$
while $i < \eta_{conv}$ **do**
    Execute rectangle and line rules ▷ new SD-model
    Evaluate SD-model
    for each `substitute` rectangle j **do**
        Obtain design response $U_{tot,i,j}$
    end for
    Cluster `substitute` rectangles by $U_{tot,i,j}$
    while $n_{subs,0} - \lceil n_{subs,0}/\eta_{conv} \rceil \cdot i < n_{subs,i}$ **do**
        Select $\mathbf{X}_{max}$, the cluster with the highest mean value
        for each rectangle j in $\mathbf{X}_{max}$ **do**
            Obtain $U_{tot,i,j}$, $U_{bend,i,j}$, $U_{norm,i,j}$, $U_{shear,i,j}$
            if $U_{tot,i,j} < \eta_{noise} \cdot U_{tot,mean,0}$ **then**
                continue
            end if
            for each c_k in $\mathbf{c}$ **do**
                if $c_k = 1$ **then**
                    if $U_{shear,i,j}/U_{tot,i,j} \geq \eta_{shear}$ **then**
                        Assign `truss` to rectangle j
                        break ▷ breaks the for loop
                    end if
                else if $c_k = 2$ **then**
                    if $U_{norm,i,j}/U_{tot,i,j} \geq \eta_{norm}$ **then**
                        Assign `beam` to rectangle j
                        break ▷ breaks the for loop
                    end if
                else if $c_k = 3$ **then**
                    if $U_{bend,i,j}/U_{tot,i,j} \geq \eta_{bend}$ **then**
                        Assign `flat shell` to rectangle j
                        break ▷ breaks the for loop
                    end if
                end if
            end for
            if j has type `substitute` **then**
                Assign `no structure` to rectangle j
            end if
        end for
    end while
    $i = i + 1$
end while

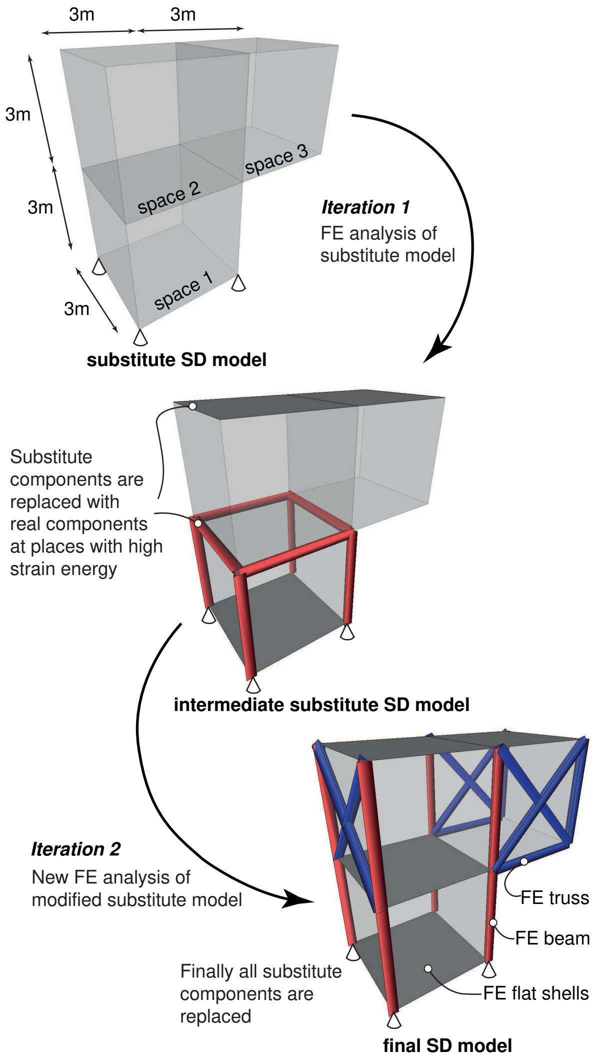

Fig. 5. Example of the iterative process of the design response grammar.

the assignment function can return any objective value that can be computed by the toolbox.

3.4.2. *Choice of Optimizer*

The multi-objective mixed-integer evolution strategy (MOMIES)—introduced in (van der Blom et al., 2019)—is used for the optimization process. This algorithm generalizes the mixed-integer evolution strategy (MIES)—described by (Li et al., 2013)—for multi-objective optimization by combining it with the multi-objective algorithm SMS-EMOA (Emmerich et al., 2005). Although in this study only categorical variables are considered, the (MO)MIES algorithm is able to optimize for problems with real, integer, and/or categorical variables. This makes it easy to extend the study to include more variables (of different types) in the future. Moreover, the algorithm employs different mutation mechanisms depending on the variable type. In this manner, it is assured that each variable type is handled appropriately.

The MOMIES algorithm is controlled by the population size μ, a reference point, and the number of function evaluations. A larger population makes it possible to maintain a more diverse set of solutions. However, it can also impede progress towards the Pareto front since it takes more time for all solutions to be updated. The reference point serves to compute the hypervolume contribution of individual solutions. The hypervolume (indicator) (Zitzler and Thiele, 1998) is the Lebesgue measure of the region covered by a set of solutions with respect to a user-defined reference point. The reference point should be dominated by all points on the Pareto front. Then, the hypervolume contribution indicates how much an individual solution contributes to the hypervolume. By comparing the hypervolume contribution of different solutions it is possible to retain the most valuable contributions. Since the reference point influences the hypervolume (and the contribution), it should be chosen carefully depending on the problem. Finally, the number of function evaluations controls how long the algorithm runs before it stops the search. A longer search may result in better solutions, but it also costs more time. Furthermore, progress may stagnate as the algorithm gets closer to the Pareto front, reducing the benefit of continuing the search for better solutions.

3.4.3. *Objectives and Constraints*

Any objective(s) that can be computed by the toolbox can be considered by the optimizer. However, the choice of objectives is problem-specific, and it is therefore considered together with the case studies, in section 4.1. No constraints are placed on the search space, although if a solution is structurally unstable, a penalty is applied to that solution.

4. Case Studies

This section presents three case studies in which the newly developed methods are investigated. In the first subsection, the settings for the methods that have been used for the case studies are presented and motivated. Thereafter, the first case study is described, in which design via optimizer assignment, and a full enumeration of the parameters of the design response grammar are applied to three archetypal building spatial designs. The performance of the design response grammar is assessed by benchmarking the results against those of the design via optimizer assignment method. Additionally, specific parameter configurations are found for which the generated layouts correspond to specific positions on the Pareto front, e.g. layouts with: minimal strain energy, minimal volume, or a balanced trade-off between these objectives (knee point). In the following subsection, the second case study is presented, in which a so-called topology optimization algorithm is modified in order to optimize the material density distribution between the components of a structural system layout. This optimization is applied on the non-dominated solutions that were found by the evolutionary algorithm in the first case study. The results show that the solutions in the Pareto front approximation retain their non-dominance (i.e. remain part of the Pareto front) after their material density distribution is optimized. If the optimization of the material density distribution (which relates to the stiffness) is regarded as a part of determining materials and dimensions in more advanced stages of the design of a system layout, this suggests that the methods produce results that are also useful in the more advanced design stages. Finally, in the third case study, a portal shaped building spatial design is subjected to design via optimizer and to the configured design response grammar using the parameter configurations that have been established in the first case study. It is then verified whether the found parameter configurations indeed lead to layouts that are located near the desired positions on the Pareto front approximation of the evolutionary algorithm.

4.1. General Settings

The settings for the presented methods that are not varied in the case studies are presented here. These settings entail the material properties, dimensions, loads, and optimization objectives. Materialization and dimensioning are not varied in the case studies, because the current work focuses on finding structural system layouts for conceptual building spatial designs. Considering such settings will increase the level of detail of a solution, and increase the size and complexity of the search space. A high level of detail in a structural system layout solution is inconsistent with the level of detail of the conceptual building spatial design for which the solution was found. Besides, an increase in the size and complexity of the search space can be handled by an evolutionary algorithm, but the parameter study for the design response grammar can quickly become computationally too expensive. Additionally, the design response grammar would require extra settings and parameters to calibrate the design rules that determine the material choice and dimensions. To that end, the presented algorithm (algorithm 1) should be extended with more rules and more design responses, which is not carried out in the presented work. This, because the focus is put on the generalizability of the solutions of the design response grammar in order to be able to quickly find structural system layouts that are sensible from a structural engineering point of view.

In this paper, two commonly used objectives for structural and topology optimization are used: (a) minimal total strain energy U[N mm], which is the sum of strain energy over all elements and all load cases in the structural model; (b) minimal total structural volume V[m^3], which is the sum of volumes of all elements in the structural model. Minimal strain energy is the governing and by far most frequently used objective in structural topology optimization, because it is yields high stiffness designs, but partly also because optimizing a system for equally distributed maximum stresses—which is more practical—proves to be very complex. The objective of minimal volume will require structures to be material efficient. Other objectives, like monetary or environmental costs, or buckling or stress constraints, could also be used, but for such objectives and constraints more specific dimensions and material selections should be included, and these are, as mentioned earlier, not considered in this paper. Also, the second case study will show that it is likely that the objectives used here are also valuable in more advanced stages of the design process.

The structural properties for the components in the structural model are all given the same generic material properties and dimensions. This is to allow a fair comparison of the objectives between different structural designs. The values of the structural properties that are used for the case studies are given in Appendix A, tables A.1 - A.4. The mesh size that is applied to the components in the structural model is $n = 3$, which has been determined based on a convergence study of some typical structural designs for the building spatial designs in this work.

Wind loading is applied in four directions, i.e. one wind load perpendicular to each orthogonal plane, where the magnitudes are simplified values similar to those found in building codes and regulations. One load case for the live load on the floors is defined, which is applied to each horizontally oriented surface of a space. The values of these loads are given in Appendix A, table A.5.

4.2. Case Study: Performance and Parameters

In this case study, the design response grammar and design via optimizer assignment are applied to three different building spatial designs. The parameters in the design response grammar are varied through a parameter study, which serves two goals. Firstly, to compare the results of the design response grammar with the global search performed by the optimizer. And secondly, to determine recommended values, or ranges thereof, for the parameters in the design response grammar.

Three building spatial designs, which are shown in figure 6, are considered in this study. More details on the spatial layout and dimensioning of these designs are given in Appendix B, figures B.1 - B.3. The considered designs are designed according to the following archetypes: A tall building with a central core, a low rise apartment building with horizontal galleries, and a large hall with possibly large spans.

The general design settings apply to both the design response grammar and design via optimizer assignment.

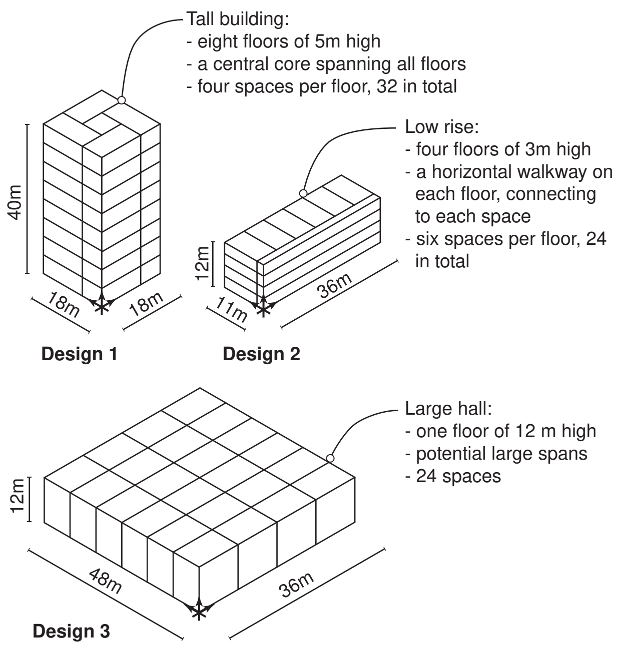

Fig. 6. The designs for the performance case study.

4.2.1. Design via Optimizer Assignment

In order to find a suitable reference point for the hypervolume (section 3.4.2) for each of the considered designs, a few trial runs have been conducted. Based on this, reference points were chosen such that they are dominated by any of the solutions observed for their corresponding design. The following values for the structural strain energy objective were determined: 2e11N m for design 1; 8e10N m for design 2; and 2e12N m for design 3. For structural volume the following values were determined: 1200m^3 for design 1; 700m^3 for design 2; and 1800m^3 for design 3. If a structural design solution is unstable, then penalty values that are equal to values of the reference point are assigned to the performances of that solution. As is standard in the mixed-integer evolution strategy, dominant crossover is used for the decision variables, while intermediate crossover is used for the step size (Li et al., 2013). A single step size is used for all decision variables, with an initial value of $1/n_d$. Here n_d denotes the number of decision variables. Further, the population size is set to $\mu = 50$ which should allow for sufficient diversity in the population considering the number of decision variables. The number of decision variables for each design are as follows: design 1: 234; design 2: 168; design 3: 106. The optimizer is given a budget of 10000 evaluations per run, and the experiments are repeated five times. This allows it to explore a reasonable part of the search space, without spending an excessive amount of time.

The results from the optimizer are shown in figure 7. On the left, for each design, all results over all runs are plotted and the non-dominated solutions of each run are highlighted. Note that solutions that are outside of the 95^{th} percentile are not shown in the figures to better visualize the results. Recall that the set of mutually non-dominated points is termed the Pareto front approximation (PFA). In the plots, these fronts show a trade-off between the two objectives. It is expected that the objective functions are conflicting since a structural design with lower volume resembles a design with less bearing components, which is less stiff, and thus results in a higher strain energy. Another observation is made in the results of design 3, where a banded structure can be observed, which could be explained by a lack of variation due to a relatively short genome in combination with the categorical nature of the design variables.

On the right of figure 7, for each design, a selection of design solutions is depicted: a solution from the knee-point region, a well-performing solution for each objective, and an arbitrarily selected poor-performing solution. Where a poor-performing solution is selected visually from the plots, from in-between 30% and 70% of their ranges. It is difficult to notice regularity in the selected designs, however, it can be noticed that for optimal stiffness in general more flat shells are assigned. For optimal volume, predominately beams are assigned, and in the knee-point region, trusses are assigned more often.

4.2.2. Parameter Study

In order to investigate if the design response grammar can find solutions that are on or close to the Pareto front, and whether specific parameter configurations correspond to certain Pareto front locations, a parameter study is performed. The settings of the design response grammar and its parameters are given first, then, in the rest of this subsection the results are presented and discussed.

Clustering in the design response grammar is performed using a minimum number of clusters $k_{min} = 4$, a maximum number of clusters $k_{max} = 10$, and a number of runs per cluster size $l = 50$. The other settings used with the design response grammar are subject to the parameter study where the parameters are investigated as follows. The thresholds for shear strain energy η_{shear}, bending strain energy η_{bend}, and normal strain energy η_{norm} are all varied from 0 to 1 with increments of 0.1 (including the boundary values 0 and 1). For the lower bound threshold of the total strain energy of a substitute component η_{noise} the values $0.025, 0.050, 0.075$ are considered. Then the threshold to control the number of iterations η_{conv} is varied from 1 to 4. Finally, all permutations of $\{1, 2, 3\}$ for the parameter of the checking order $\mathbf{c}$ are tested. These variations in the settings result in 95,832 different parameter configurations that are together totaling 383,328 finite element simulations. Each configuration is evaluated for each design.

In figure 8, on the left, the results of the parameter study are given, together with the overall Pareto front approximation that was obtained from the optimizer. On a first note, it should be mentioned that not every combination of parameters has resulted in a performance in these plots, because unstable structural design models have been disregarded for this study (on average 24.6% is disregarded). On a second note, the dashed boxes around the PFAs are the selection of solutions that will be used for an analysis of the parameter study which follows later. On a third and final note, the axis for the strain energy has been scaled on a log scale with the purpose to better visualize the results, unlike the plots in figure 7. Compared to the results of the parameter study, the EA achieves better coverage of the knee point region, whereas the design response grammar found new non-dominated solutions in the extremal regions. Nevertheless, the parameter study also found non-dominated solutions close to the Pareto front approximation. Altogether, these results show that the design response grammar can generate qualitatively good solutions.

On the right of figure 8, for each design, a selection of the solutions found by the design response grammar are shown. This selection contains a solution from the knee-point region, a well-performing solution for each objective individually, and an arbitrarily selected poor-performing solution. From this selection, it can be noticed that a design with many trusses will lead to a design in the knee-point region, a design with many flat shells to a stiff design, and a design with many beams to a material efficient design. It should be noted that flat shell elements on the façade obstruct the view of possible internal structure, separate designs in figure 8 (i.e. 10 and 12) may for that reason appear similar.

A recommendation of a parameter configuration that will yield solutions that perform well for the objectives is essential for the design response grammar to be useful. Because a full enumeration of parameters is too expensive to repeat with new design tasks, for that matter using design via optimizer assignment would be a more fitting choice because it uses fewer

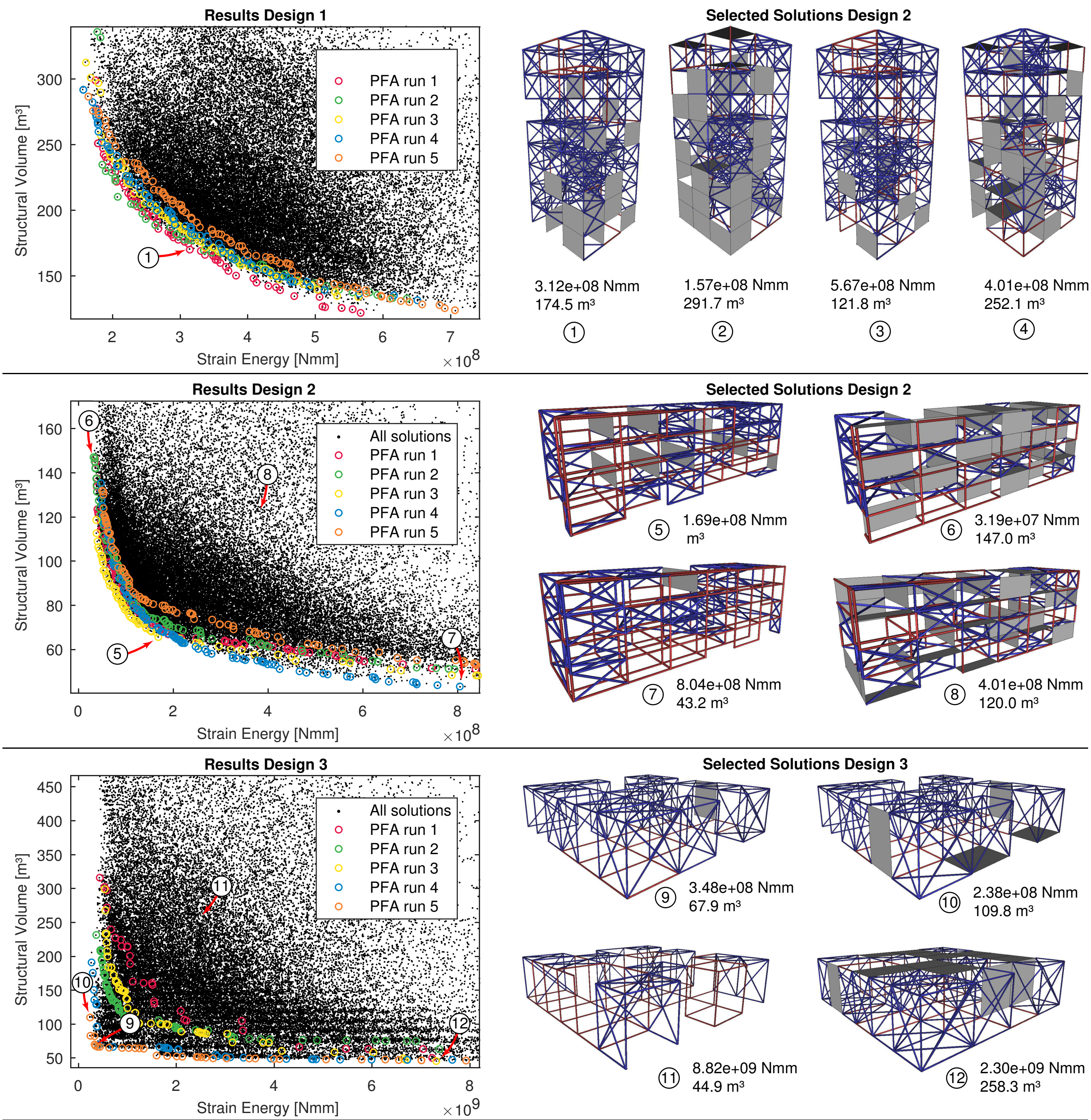

Fig. 7. Results of design via optimizer assignment, performance plots and visualized solutions.

evaluations and achieves better coverage in the knee-point area. In the next part, it is therefore studied if solutions of the design response grammar at specific locations of the Pareto front approximations can be expressed in terms of parameters configurations. This is investigated here using parallel coordinate plots as depicted in figure 9. In the parallel coordinate plots, each design that results from a combination of parameters is represented by a polyline that is plotted from axis to axis. The first two axes show the performances of the designs, and the rest of the axes represent the parameters. Plotted in grey are all considered solutions and plotted in each color is a different selection of solutions, where the colored dashed line indicates the bounds of this selection.

Looking at the plots, it can be observed from the blue lines (designs within the blue dashed box in figure 8) that η_{shear} is always zero, but none of the other parameters show a clear correlation. So also no clear recommendation can be given based on this selection alone. Reducing the upper bound and the right bound of the box around the Pareto front approximation yields the designs highlighted in red (designs within the red dashed

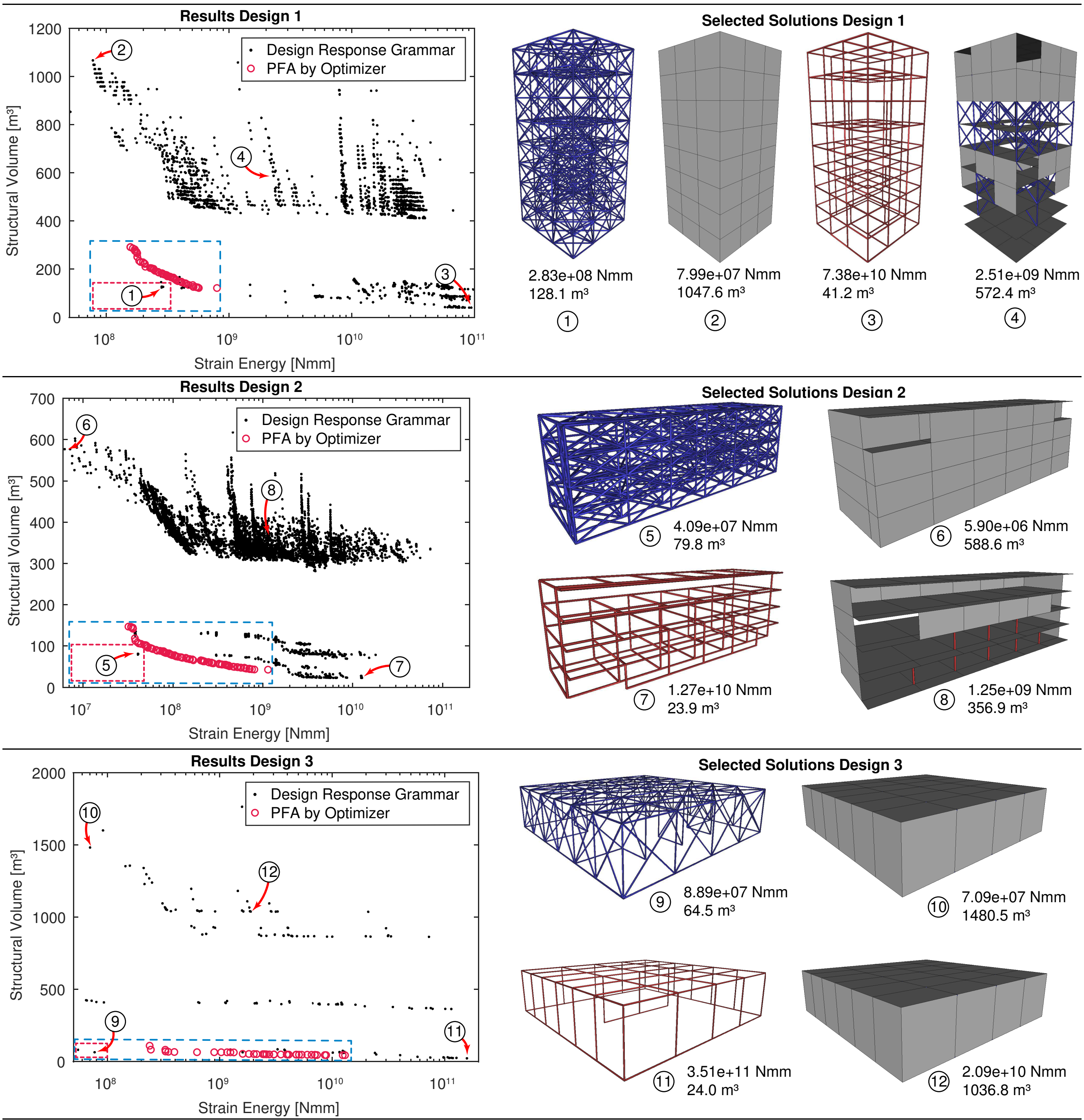

Fig. 8. Comparison plots of design via optimizer assignment and design response grammar.

box in figure 8). However, for the designs that remain, a correlation between parameters can still not be observed. Therefore, the selection is further reduced to show only designs that result from checking orders of 2, 3, 1 and 3, 2, 1. The rationale behind this reduction is the fact that the threshold for shear strain energy (check 1) is zero, and assessing this check first will thus always result in an assignment of the `truss` type. Checking for shear strain energy (check 1) last gives a chance for the other types of strain energy (checked by check 2 and check 3) to be assigned. In the case studies, however, it was not observed that these checks result in a type assignment. This can be explained from the fact that for the lines plotted in red the thresholds for checks 2 and 3 are high (0.9 and 1.0). With regards to the convergence parameter η_{conv}, it is concluded that one iteration is sufficient. A value of 2.5% is recommended for the noise threshold based on the results of design 3. In summary, for each evaluated design the knee point solution can be generalized into the following parameter configuration: $\eta_{shear} = 0$, $\eta_{norm} = 1.0$, $\eta_{bend} = 1.0$, $\eta_{conv} = 1$, $\eta_{noise} = 0.025$, and finally $\mathbf{c} = 2, 3, 1$ or $3, 2, 1$.

By interpreting this generalization of the parameters, it be-

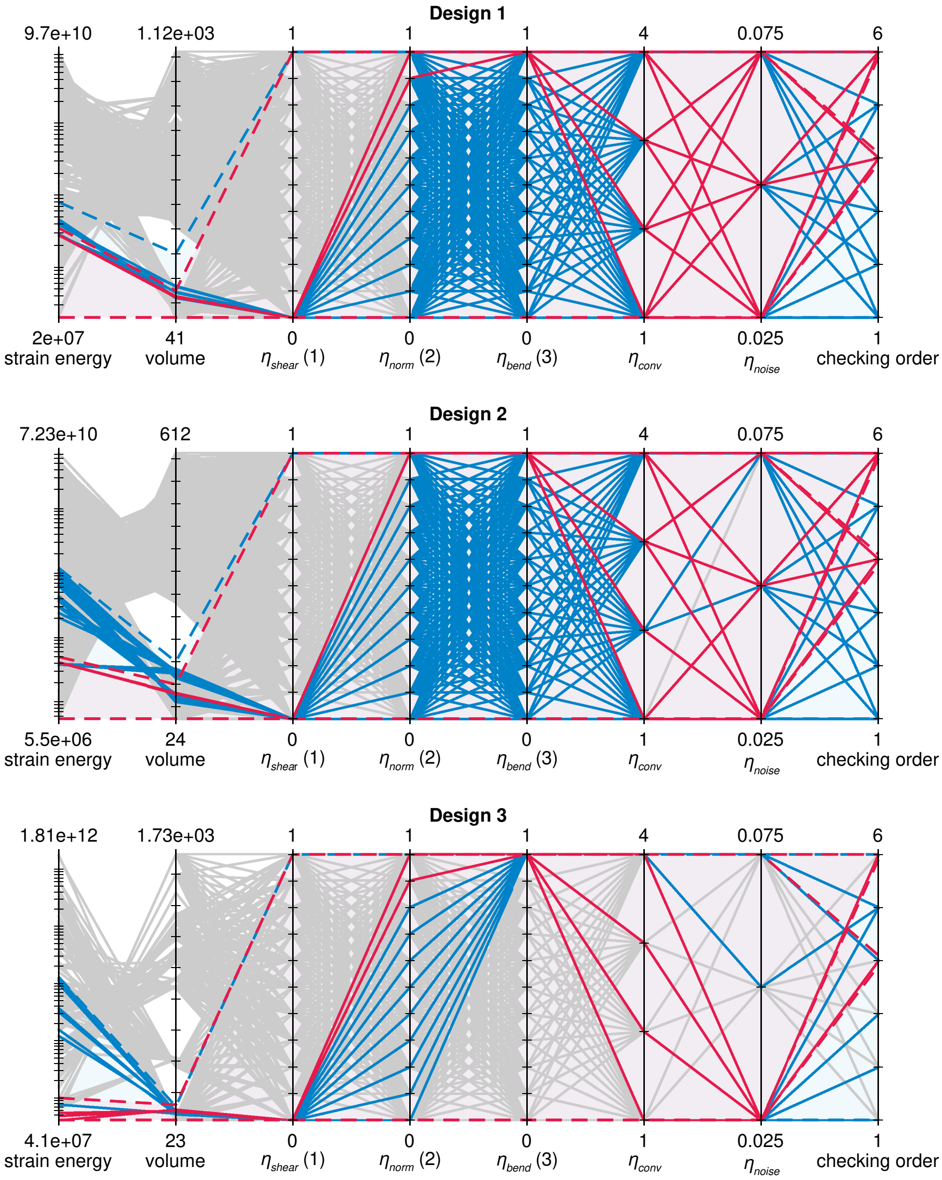

Fig. 9. Parallel coordinate plots of the performances and the parameters in the parameter study.

comes apparent that an all truss structural design will always be the result of the design response grammar. This can be explained from the fact that a combination of diagonal trusses is a well-known stabilization method that does not require much material. Moreover, this may well be the reason that not more than one iteration is required within the grammar. In a similar fashion as described before, a different study using parallel coordinate plots has been performed. For brevity, these plots are not presented in this work, however, the found parameters and the search methodology are presented in the following. The study has been performed such that the higher strain energies that lie outside of the box around the Pareto front approximation of figure 8 are considered as well. The study shows that η_{shear} can then also hold values of 0.1 or 0.2. By restricting the parallel coordinate plots to only these values for η_{shear}, it has been concluded that the value for η_{bend} can be fixed at 1.0 and that of η_{norm} should now be set to zero. Values for η_{conv} are still recommended to be set to 1 iteration, and for η_{noise} the recommendation is still 0.025. Regarding the checking order, the check for bending strain energy appears to be irrelevant, and the check for shear strain energy should precede that of bending. A fitting checking order would then be 1, 2, 3. Using these parameter configurations, the `beam` type will be assigned by the grammar more often.

It is desirable that the grammar can explore the complete Pareto front, and as such would also be able to discover designs with `flat shell` assignments. These assignments would provide more stiffness at the cost of structural volume. However, similar studies like above, using parallel coordinate plots, did not yield any parameter configuration for solutions that have a low amount of strain energy and have the `flat shell` type assigned more often. The cause may be observed in figure 8, where large horizontal bands without results can be seen. From the optimizer's results in figure 7 it is clear that solutions do exist within these bands. It appears that these banded gaps are caused by the discretization of η_{shear} and η_{bend}. A refinement of the discretization of parameters may thus improve the results of the parameter study, and it cannot yet be concluded that the design response grammar cannot explore the complete Pareto front. Such a refinement in the parameter study is not performed here, because the focus is put on the generalizability of the solutions.

4.3. Case study: Optimal Material Density Distribution

In the presented work, generic materials and dimensions are assigned to the generated structural system layouts. Although this simplifies the design problem as well as the generation and assessment of layouts, the resulting layouts appear to be impractical regarding aspects like dimension-to-span ratios or stress constraints. To investigate whether the generated structural system layouts are still useful in more advanced stages of the design process, this second case study is presented. The study applies a technique similar to topology optimization to optimize the material density distribution of each individual structural component. By applying this optimization technique on structural system layouts that are part of the Pareto front approximations as found by the evolutionary algorithm in the first case study, it is shown that—after optimization—the fronts remain the same qualitatively. In other words, the performance trade off between two layouts, which can be deduced from the Pareto front approximation, still holds after their stiffness distribution is optimized. If optimization of the material density distribution (more or less equivalent to an optimization of the stiffness distribution), is regarded as a part of determining materials and dimensions in more advanced stages of the design of a system layout, this suggests that the methods produce results that are also useful in the more advanced design stages.

4.3.1. Algorithm

Optimization of the material density distribution is applied in practice to find the stiffest structure within a continuum design domain, given certain loading conditions and a material constraint, this is often referred to as topology optimization (Bendsøe and Sigmund, 2004). In this subsection, modifications to a topology optimization algorithm are introduced, such that the algorithm considers the densities of different element types separately, with a single density for all elements within a structural component. The problem formulation of the original topology optimisation algorithm (Andreassen et al., 2011) is given in equation 9. Here $\mathbf{x}$ denotes the vector holding the density x_e of each element e, V_0 is the sum of all element volumes: $\sum_{e=1}^{N} V_{0,e}$, whereas $V(\mathbf{x})$ is denoted as $\sum_{e=1}^{N} x_e V_{0,e}$, the objective c is called compliance, p is a penalty factor to push element densities more towards either zero or one, and N is the number of elements. The element densities modify the stiffness of the elements in the FE model, as such, a redistribution of element densities signifies a redistribution of the material and stiffness.

$$
\begin{aligned}
\min_{\mathbf{x}}:\ & c(\mathbf{x}) = \sum_{e=1}^{N} x_e^p \mathbf{u}_e^{\intercal} \mathbf{K}_e \mathbf{u}_e \\
\text{subject to:}\ & V(\mathbf{x})/V_0 = f \\
& \mathbf{f} = \mathbf{K}\mathbf{u} \\
& 0 \le \forall \in \mathbf{x} \le 1
\end{aligned}
\tag{9}
$$

The algorithm uses a gradient of the objective function with respect to changes in element volume and changes in element density. The gradient is filtered to make the solution less mesh dependent, this will, for example, prevent checkerboard patterns. Finally, a bi-sectioning algorithm is used to update the densities using the filtered gradient while satisfying the volume and density constraints. This process is iterated until the greatest change in element densities is less than a set threshold. More details on the implementation can be found in Andreassen et al. (2011).

The above topology optimization algorithm does not distinguish in element types, e.g. truss, beam, or flat shell. The algorithm could therefore distribute (all) the density of one element type to elements of other types. However, this is not desirable when the algorithm is used to assess a structural system layout in which it is precisely the composition of different element types that is of importance. Therefore a modified volume fraction constraint is introduced in equation 10.

$$
V_i(\mathbf{x}_i)/V_{i,0} = f \tag{10}
$$

In the new problem formulation, i denotes the type of element (e.g. truss, beam, or flat shell). Filtering of the gradient is then also performed separately per element type to prevent an influence on the density of one type of element by elements of a different type. Moreover, the number of volume constraints has increased as a consequence of the modified problem formulation. This also increases the number of times the bi-sectioning algorithm should be executed, once for each element type i.

Finally, the algorithm will in this work be used as a post-processing step to find optimal stiffness distributions between structural components. Stiffness variations within components are not considered, therefore the densities are optimized component-wise rather than element-wise. Note that no mesh dependency filter needs to be used when the densities are varied component-wise, because the problem is no longer mesh dependent.

4.3.2. Settings

Optimization of the material density distribution is performed for each design with the following settings. The penalty factor is set to $p = 3.0$. Three different values for the volume fraction f are used: 0.2, 0.5, and 0.8. Therefore, three different runs of the material density distribution optimization algorithm are carried out for each design. And finally, the stopping criterion has been set to stop with absolute density changes smaller than 0.01.

4.3.3. Results

For each building spatial design the results are plotted in figure 10. The Pareto front approximation, as found in section 4.2 is plotted with black circles, note that a volume fraction $f = 1.0$ will lead to a solution equivalent to the original, because no redistribution of material density can take place. Plotted in color, are the different volume fractions used for the optimizations, circles represent solutions with a uniformly distributed material density (i.e. $\forall e : x_e = f$) and crosses represent solutions after their material density distribution has been optimized. Note that for each design, the initial structural volume (V_0) remains the same, for different volume fractions a point representing a solution thus only moves horizontally. When observing the plots regarding optimality, it can be noticed that each set of optimized designs still forms a set of non-dominated solutions.

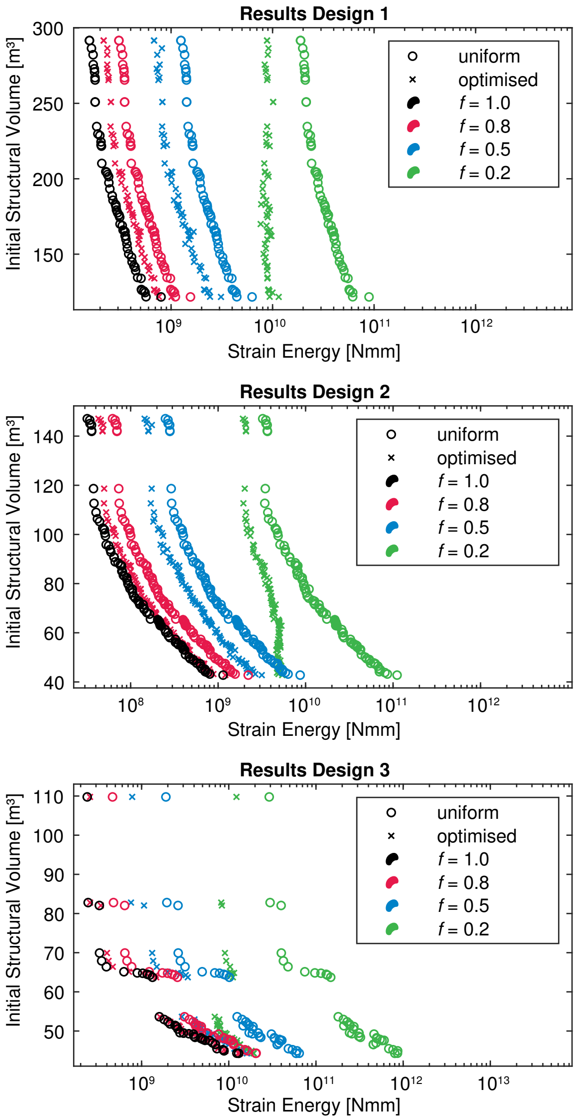

Fig. 10. Results of topology optimization on non-dominated solutions from figure 7

4.4. Case Study: Comparison and Validation

The case study in this section is intended to compare and validate the pre-configured design response grammar against design via optimizer assignment. For this purpose, a new building spatial design is introduced, see figure 11, and for more details see figure B.4 in Appendix B. Structural designs for this building spatial design have been created using both methods. Settings for the optimizer are the same as presented in section 4.1. For the design response grammar, the used parameter configurations are summarized in table 2. These are parameter configurations that have been found in the first case study (section 4.2.2), which are expected to yield structural system layouts that are located near specific points on the Pareto front. Other settings for the design response grammar are the same as the settings presented in section 4.1.

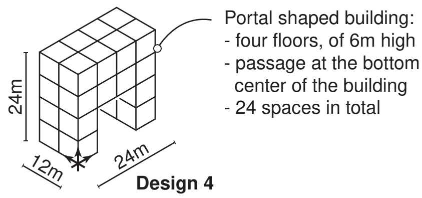

Fig. 11. Design 4, a portal shaped building.

Table 2. Parameter configurations for the design response grammar.

id	η_{shear}	η_{norm}	η_{bend}	η_{conv}	η_{noise}	**c**
1	0.0	1.0	1.0	1	0.025	{3, 2, 1}
2	0.1	0.0	1.0	1	0.025	{1, 2, 3}
3	0.2	0.0	1.0	1	0.025	{1, 2, 3}

4.4.1. Results

Plotted together in figure 12 are the results from design via optimizer assignment and the design response grammar. All solutions found by design optimizer assignment are plotted as black dots, and non-dominated points that form the Pareto front approximation are highlighted with red circles. The Pareto front approximation dominates the solutions found by the design response grammar, indicating design via optimizer assignment finds better solutions than the design response grammar. Moreover, the evolutionary algorithm found a more evenly distributed Pareto front approximation, which can yield more information regarding trade-offs. Nevertheless, the design response grammar resulted in solutions that are close to the desired points on the Pareto front approximation: configuration 1 is located near the knee-point, and configurations 2 and 3, respectively, are more optimal regarding the volume objective. This suggests that the configurations found by the parameter study do generalize to other building spatial designs.

Regarding computational cost, the three layouts found by the design response grammar were found after six evaluations (two for each solution, once the substitute model and once the final design). Whereas, the Pareto front approximation was obtained after 50000 evaluations. The found generality thus allows the

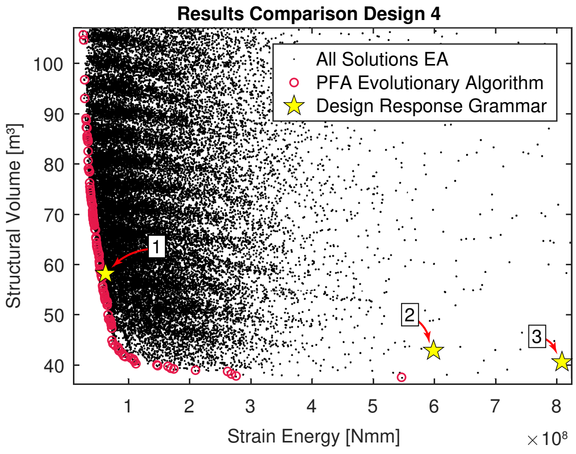

Fig. 12. Performance plot of the structural design solutions for design 4.

grammar to find solutions that perform well in both objectives without repeating an extensive parameter study, while having a much lower computational cost (50000 vs. 6 evaluations). However, based on the results of this case study alone, no proof or guarantee regarding the extent to which solutions generalize—and thus the optimality of the found solutions—can be given for the design response grammar.

Another aspect to consider in the comparison between the two methodologies is the required input. Design by optimizer assignment requires the user to define input that is related to structural design (e.g. ranges of materials, dimensions, connections, and loads), but also settings related to the optimizer like population size, reference point, and an evaluation budget. Whereas the parameters of the design response grammar are directly related to phenomena within the field of structural design. A structural design engineer is thus likely to have a better understanding of the required input for the design response grammar, which is in general considered desirable for the application of a method.

Finally, the structural design solutions that were generated by the design response grammar are depicted in figure 13. From the solutions, it can be observed that parameter configuration 1 leads to a full truss design. Whereas, the other two configurations, which were defined to assign the `beam` type more often, indeed generate solutions with fewer trusses at locations where they are not effective.

5. Discussion

In this paper two newly developed design methods are presented. The first, the design response grammar, uses design rules—configurable by parameters—to develop a structural system layout step by step as a function of a building spatial design's geometry and preliminary assessments of the structural system under development. The second, design via optimizer assignment, uses an evolutionary algorithm to find a Pareto front approximation of structural system layouts for a conceptual building spatial design. In this section, some critical remarks on the developed methods and the presented work are given.

Firstly, the number of structural types that can be assigned to the geometry of the building spatial design is limited. This prohibits the design of more complex, but common, structural systems such as floor slabs supported by beams. However, the presented method can be extended to support the assignment of more structural types to a greater variety of geometries. In that case, the assignment criteria based on design responses have to be reconsidered.

Moreover, the assignment of structural types based on a single type of load within the design response may not be adequate. For example, in this paper a flat shell is only assigned when out-of-plane strain energies are high, while it is also excellent in dealing with in-plane forces. In that case, a structural type like a space truss might be more suitable when only out-of-plane strain energy is present, and a flat shell when a combination of out-of-plane and in-plane strain energies is present.

Solutions that perform well for the objectives in this work are most often trussed designs. This can be due to the choice of objectives, i.e. a design must be material-efficient (minimal volume) and it must be stiff (minimal strain energy). Other objectives like cost, construct-ability, and practicality have not yet been considered. Accounting for only two objectives, the design response grammar—in its current form—may not yet be suitable for its intended purposes within a framework of more general building spatial design optimization. Moreover, with other objectives, finding parameter configurations that correspond to certain desirable locations on the Pareto front may be less straightforward, or perhaps not even possible.

In practice, also constraints like a maximum allowable stress or buckling avoidance determine the feasibility of a design. These constraints affect the search space, e.g. a maximum achievable span of a structural type due to an exponentially increasing self-weight. To take into account such constraints the considered solutions should have realistic material properties and dimensions, which is not the case in the presented work. Materialization and dimensioning are left out of consideration in the presented work because the focus is put on the quick generation of structural system layouts for conceptual building spatial design. Introducing materials and dimensions as design variables would make the search space larger and the design response grammar more complex. Moreover, it would introduce a discrepancy between the level of detail of the building spatial design and the generated structural system layouts. Nevertheless, in future work, rules of thumb that limit each type of structure to a certain span range (e.g. in practice a common maximum span for a monolithic floor structure is 7 m) could ensure the feasibility of the solutions that are found by the presented methodologies. Specifically with respect to stress-based design, topology optimization is very complex, and no single robust method for this is available yet. Promising is proportional topology optimization (Biyikli and To, 2015). However, this is not considered in the current work, and the presented methods should thus be extended in this regard before being applied in practice.

The second case study has been used to predict that the layouts found by the current methods and objectives are also useful in more advanced stages of the design process. However, a next

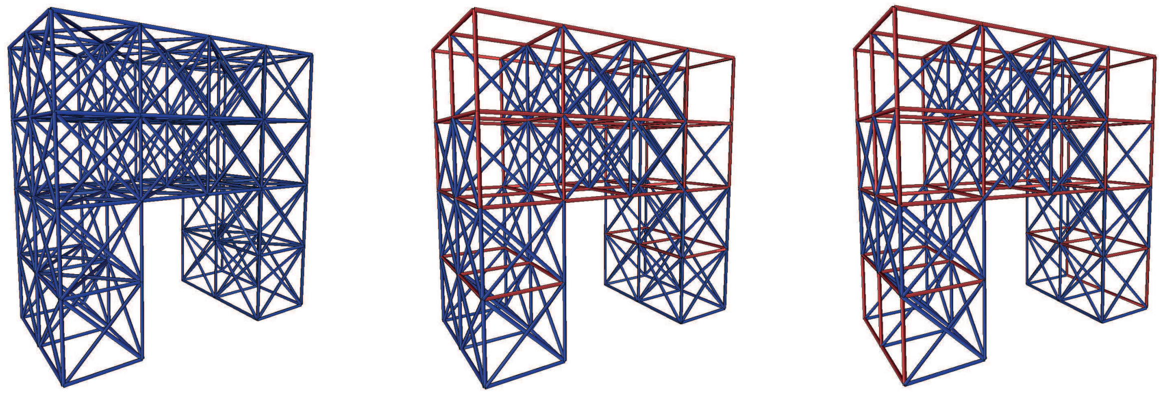

Fig. 13. Design response grammar solutions for design 4.

stage in the design process could be to scale all material density distributions (more or less equivalent to stiffness) uniformly up or down to achieve a certain maximum allowable displacement or stress in the structural system layout. As the required scale factor will be different for each layout, it follows that each layout will acquire different new values for structural volume and strain energy. So the qualitative character of the Pareto front that remained the same after topology optimization, may not remain the same after the material density distributions are scaled for displacements or stresses. A final answer to the importance of this issue can only be given when element dimensions and materials are included in the methods in a detailed fashion.

Finally, the parameters in the design response grammar have been selected and configured through insight into the problem. The problem is however complex, and other techniques to identify and configure parameters may yield better results. For example, machine learning may be used, where the substitute structural design model serves as input, a structural design solution or a collection of non-dominated structural design solutions are output, and non-dominated solutions are used as training data. Also other data learning techniques—e.g. innovization (Deb and Srinivasan, 2006)—can be applied to the optimization results to find relationships between the features of a substitute model and the optimization results of an evolutionary algorithm. Additionally, in the parameter study, the discretization of continuous parameters led to large gaps in the non-linear relationships between parameters and the objectives. Parameter tuning using, for instance, gradient-based techniques will give more insight into these relationships. The latter may be carried out with an optimizer, which may even reduce the computational cost of the parameter study as optimizers are designed to avoid full enumeration.

The above remarks show that future work on the design response grammar is required. However, this does not discount the potential of the design response grammar that has already been observed. For the common objectives of minimal strain energy and minimal structural volume, and after the performed parameter study, it was possible to find parameter configurations for the design response grammar that yield structural system layouts that perform such that they are located at desirable positions on the Pareto front found by an evolutionary algorithm. It is very likely that, by generalizing these results for other building spatial designs, specific points on the Pareto front approximation can be expressed in terms of parameter configurations. In doing so the grammar can accurately find near Pareto optimal design solutions with considerably less effort than an optimization algorithm (6 vs 50000 evaluations). A method that can quickly generate solutions that perform well for the defined objectives for conceptual building spatial designs is thus found to be a realistic goal.

6. Conclusions and Outlook

Motivated by the application in (multi-disciplinary) building spatial design optimization, an existing optimization toolbox has been extended with two new methods that can find structural system layouts for a building spatial design that perform well for a given set of objectives. The first, the design response grammar, uses design rules—configurable by parameters—to develop a structural system layout step by step as a function of a building spatial design's geometry and preliminary assessments of the structural system under development. The second, design via optimizer assignment, uses an optimizer to determine the placement of structural components in structural models.

Both methods can find solutions that perform well for objectives that are commonly used for structural and topology optimization: minimal strain energy and minimal structural volume. The design via optimizer assignment method yields evenly distributed Pareto front approximations, from which insight into the trade-off between objectives can be gained.

Through a parameter study, it has been demonstrated that specific parameter configurations of the design response grammar lead to specific desirable locations on the the Pareto front approximation that was found by the optimizer. By generalizing, these specific points on the Pareto front approximation can be expressed in terms of parameter configurations. This reduces the computational cost significantly compared to design via optimizer assignment, making the design response grammar useful for cases where many different or rapidly evolving building spatial designs should be assessed for their structural design potential.

In the presented work, typical objectives for structural optimization were used: minimal strain energy and minimal volume. These objectives allow for leaving out detailed materialization and dimensioning, which: (1) reduces the size and complexity of the search space; and (2) avoids a discrepancy between the level of detail of a conceptual building spatial design and the structural system layout. Naturally, generic material properties and dimensions still need to be used, but as a consequence, practical constraints like allowable stress, buckling, or deformation are not useful to be checked.

This paper also presented an optimization technique similar to topology optimization to optimize the material density distributions of each individual structural component, which can be regarded as a part of determining materials and dimensions in more advanced stages of the design of a system layout. This technique was applied to the layouts that are part of the Pareto front approximations as found by the evolutionary algorithm in the first case study, it was shown that—after optimization—the fronts remain the same qualitatively, which suggests that the methods produce results that are also useful in more advanced design stages.

Finally, critical remarks regarding the design variables, design response, objectives, constraints, and parameter study have been made, and it is clear that the design response grammar needs to be developed and validated further. Future research should involve the development of additional structural element types for the design response grammar to increase the variety of possible solutions; the exploration of new objectives and constraints to further increase the feasibility of the layouts; the investigation of state-of-the art techniques like machine learning in the assignment of structural types based on the mechanical response to avoid complex assignment rules and to possibly improve the results.

Acknowledgements

This work is part of the TTW-Open Technology Program with project number 13596, which is (partly) financed by the Netherlands Organization for Scientific Research (NWO). The authors acknowledge Niels ten Heggeler for his ideas and research that sparked the development of the design response grammar. Moreover, the authors wish to express their gratitude towards Rob Claessens for his research on the design response grammar, and to Thijs de Goede for his contributions to the toolbox.

References

Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B.S., Sigmund, O., 2011. Efficient topology optimization in matlab using 88 lines of code. Structural and Multidisciplinary Optimization 43, 1–16. doi:10.1007/s00158-010-0594-7.

Azzouz, A., Borchers, M., Moreira, J., Mavrogianni, A., 2017. Life cycle assessment of energy conservation measures during early stage office building design: A case study in London, UK. Energy and Buildings 139, 547–568. doi:10.1016/j.enbuild.2016.12.089.

Basbagill, J.P., Flager, F.L., Lepech, M., 2014. A multi-objective feedback approach for evaluating sequential conceptual building design decisions. Automation in Construction 45, 136–150. doi:10.1016/j.autcon.2014.04.015.

Beghini, L.L., Beghini, A., Katz, N., Baker, W.F., Paulino, G.H., 2014. Connecting architecture and engineering through structural topology optimization. Engineering Structures 59, 716–726. doi:10.1016/j.engstruct.2013.10.032.

Bendsøe, M., Sigmund, O., 2004. Topology Optimization. Theory, Methods, and Applications. second ed., Springer. ISBN: 978-3-662-05086-6.

Biyikli, E., To, A.C., 2015. Proportional topology optimization: a new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in matlab. PLOS ONE 10, 1–23. doi:10.1371/journal.pone.0145041.

van der Blom, K., Boonstra, S., Hofmeyer, H., Bäck, T., Emmerich, M.T., 2017. Configuring advanced evolutionary algorithms for multicriteria building spatial design optimisation, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE. pp. 1803–1810. doi:10.1109/CEC.2017.7969520.

van der Blom, K., Boonstra, S., Hofmeyer, H., Emmerich, M.T.M., 2016a. Multicriteria building spatial design with mixed integer evolutionary algorithms, in: Handl, J., et al. (Eds.), Parallel Problem Solving from Nature – PPSN XIV, Springer International Publishing. pp. 453–462. doi:10.1007/978-3-319-45823-6_42.

van der Blom, K., Boonstra, S., Hofmeyer, H., Emmerich, M.T.M., 2016b. A super-structure based optimisation approach for building spatial designs, in: Plevris, V., Papadrakakis, M., Papadopoulos, V., Stefanou, G. (Eds.), Congress proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS, pp. 3409–3422. doi:10.7712/100016.2044.10063.

van der Blom, K., Yang, K., Bäck, T., Emmerich, M.T.M., 2019. Towards multi-objective mixed integer evolution strategies, in: LeGO 2018 – Proceedings of the Int. Workshop on Global Optimization, pp. 1–4. doi:10.1063/1.5090013.

Boonstra, S., van der Blom, K., Hofmeyer, H., Emmerich, M.T.M., 2017. Combined super-structured and super-structure free optimisation of building spatial designs, in: Koch, C., Tizani, W., Ninic, J. (Eds.), 24th EG-ICE International Workshop on Intelligent Computing in Engineering, University of Nottingham, United Kingdom. pp. 23–34. ISBN: 978-1-5108-4345-5.

Boonstra, S., van der Blom, K., Hofmeyer, H., Emmerich, M.T.M., van Schijndel, J., de Wilde, P., 2018. Toolbox for super-structured and super-structure free multi-disciplinary building spatial design optimisation. Advanced Engineering Informatics 36, 86–100. doi:10.1016/j.aei.2018.01.003.

Brown, N.C., Mueller, C.T., 2016. Design for structural and energy performance of long span buildings using geometric multi-objective optimization. Energy and Buildings 127, 748–761. doi:10.1016/j.enbuild.2016.05.090.

Chong, Y.T., Chen, C.H., Leong, K.F., 2009. A heuristic-based approach to conceptual design. Research in Engineering Design 20, 97–116. doi:10.1007/s00163-008-0059-9.

Clevenger, C.M., Haymaker, J., 2011. Metrics to assess design guidance. Design Studies 32, 431–456. doi:10.1016/j.destud.2011.02.001.

Clevenger, C.M., Haymaker, J.R., 2012. The value of design strategies applied to energy efficiency. Smart and Sustainable Built Environment 1, 222–240. doi:10.1108/20466091211287119.

Deb, K., Srinivasan, A., 2006. Innovization: Innovating design principles through optimization, in: GECCO '06 Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA. pp. 1629–1636. doi:10.1145/1143997.1144266.

Emmerich, M., Beume, N., Naujoks, B., 2005. An EMO algorithm using the hypervolume measure as selection criterion, in: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 62–76. doi:10.1007/978-3-540-31880-4_5.

Emmerich, M.T.M., Deutz, A.H., 2018. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural Computing 17, 585–609. doi:10.1007/s11047-018-9685-y.

Fenves, S.J., Rivard, H., Gomez, N., 2000. SEED-Config: a tool for conceptual structural design in a collaborative building design environment. Artificial Intelligence in Engineering 14, 233–247. doi:10.1016/S0954-1810(00)00018-2.

Gerber, D.J., Lin, S.H.E., 2014. Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. Simulation

90, 936–959. doi:10.1177/0037549713482027.

Gero, J.S., D'Cruz, N., Radford, A.D., 1983. Energy in context: a multicriteria model for building design. Building and Environment 18, 99–107. doi:10.1016/0360-1323(83)90001-X.

Geyer, P., 2009. Component-oriented decomposition for multidisciplinary design optimization in building design. Advanced Engineering Informatics 23, 12–31. doi:10.1016/j.aei.2008.06.008.

Geyer, P., Schlueter, A., 2014. Automated metamodel generation for design space exploration and decision-making–a novel method supporting performance-oriented building design and retrofitting. Applied Energy 119, 537–556. doi:10.1016/j.apenergy.2013.12.064.

Guennebaud, G., Jacob, B., et al., 2019. Eigen v3: a C++ linear algebra library. URL http://eigen.tuxfamily.org.

Hamdy, M., Nguyen, A.T., Hensen, J.L., 2016. A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. Energy and Buildings 121, 57–71. doi:10.1016/j.enbuild.2016.03.035.

Haymaker, J., Kunz, J., Suter, B., Fischer, M., 2004. Perspectors: composable, reusable reasoning modules to construct an engineering view from other engineering views. Advanced Engineering Informatics 18, 49–67. doi:10.1016/j.aei.2004.10.002.

Hofmeyer, H., Davila Delgado, J.M., 2015. Coevolutionary and genetic algorithm based building spatial and structural design. AI EDAM 29, 351–370. doi:10.1017/S0890060415000384.

Hopfe, C.J., Emmerich, M.T., Marijt, R., Hensen, J., 2012. Robust multi-criteria design optimisation in building design. Proceedings of the First Building Simulation and Optimization Conference, Loughborough, UK , 118–125.

Hopfe, C.J., Hensen, J.L., 2011. Uncertainty analysis in building performance simulation for design support. Energy and Buildings 43, 2798–2805. doi:10.1016/j.enbuild.2011.06.034.

Khemlani, L., Timerman, A., Bennen, B., Kalay, Y.E., 1998. Intelligent representation for computer-aided building design. Automation in Construction 8, 49–71. doi:10.1016/S0926-5805(98)00065-X.

Li, R., Emmerich, M.T., Eggermont, J., Bäck, T., Schütz, M., Dijkstra, J., Reiber, J.H., 2013. Mixed integer evolution strategies for parameter optimization. Evolutionary Computation 21, 29–64. doi:10.1162/EVCO_a_00059.

Liggett, R.S., 2000. Automated facilities layout: past, present and future. Automation in Construction 9, 197–215. doi:10.1016/S0926-5805(99)00005-9.

Machairas, V., Tsangrassoulis, A., Axarli, K., 2014. Algorithms for optimization of building design: A review. Renewable and Sustainable Energy Reviews 31, 101–112. doi:10.1016/j.rser.2013.11.036.

MacKay, D.J.C., 2003. Information theory, inference and learning algorithms. Cambridge University Press. ISBN: 978-0-521-64298-9.

Maher, M., Tang, H.H., 2003. Co-evolution as a computational and cognitive model of design. Research in Engineering Design 14, 47–64. doi:10.1007/s00163-002-0016-y.

Mora, R., Bédard, C., Rivard, H., 2008. A geometric modelling framework for conceptual structural design from early digital architectural models. Advanced Engineering Informatics 22, 254–270. doi:10.1016/j.aei.2007.03.003.

Negendahl, K., Nielsen, T.R., 2015. Building energy optimization in the early design stages: A simplified method. Energy and Buildings 105, 88–99. doi:10.1016/j.enbuild.2015.06.087.

Nielsen, A.N., Jensen, R.L., Larsen, T.S., Nissen, S.B., 2016. Early stage decision support for sustainable building renovation–a review. Building and Environment 103, 165–181. doi:10.1016/j.buildenv.2016.04.009.

Oh, J., Hong, T., Kim, H., An, J., Jeong, K., Koo, C., 2017. Advanced strategies for net-zero energy building: Focused on the early phase and usage phase of a building's life cycle. Sustainability 9, 2272. doi:10.3390/su9122272.

Okudan, G.E., Tauhid, S., 2008. Concept selection methods–a literature review from 1980 to 2008. International Journal of Design Engineering 1, 243–277. doi:10.1504/IJDE.2008.023764.

Picco, M., Lollini, R., Marengo, M., 2014. Towards energy performance evaluation in early stage building design: A simplification methodology for commercial building models. Energy and Buildings 76, 497–505. doi:10.1016/j.enbuild.2014.03.016.

Rafiq, M., MacLeod, I., 1988. Automatic structural component definition from a spatial geometry model. Engineering Structures 10, 37–40. doi:10.1016/0141-0296(88)90014-4.

Ritter, F., Geyer, P., Borrmann, A., 2015. Simulation-based decision-making in early design stages, in: 32nd CIB W78 Conference, Eindhoven, The Netherlands, pp. 27–29.

Ruiz-Montiel, M., Belmonte, M.V., Boned, J., Mandow, L., Millán, E., Badillo, A.R., Pérez-de-la Cruz, J.L., 2014. Layered shape grammars. Computer-Aided Design 56, 104–119. doi:10.1016/j.cad.2014.06.012.

Schlueter, A., Geyer, P., 2018. Linking BIM and design of experiments to balance architectural and technical design factors for energy performance. Automation in Construction 86, 33–43. doi:10.1016/j.autcon.2017.10.021.

Sharafi, P., Samali, B., Ronagh, H., Ghodrat, M., 2017. Automated spatial design of multi-story modular buildings using a unified matrix method. Automation in Construction 82, 31–42. doi:10.1016/j.autcon.2017.06.025.

Song, H., Ghaboussi, J., Kwon, T.H., 2016. Architectural design of apartment buildings using the implicit redundant representation genetic algorithm. Automation in Construction 72, 166–173. doi:10.1016/j.autcon.2016.09.001.

Steiner, B., Mousavian, E., Saradj, F.M., Wimmer, M., Musialski, P., 2017. Integrated structural–architectural design for interactive planning. Computer Graphics Forum 36, 80–94. doi:10.1111/cgf.12996.

Stiny, G., 2006. Shape: talking about seeing and doing. MIT Press. ISBN: 978-0-262-19531-7.

Touloupaki, E., Theodosiou, T., 2017. Performance simulation integrated in parametric 3d modeling as a method for early stage design optimization—a review. Energies 10, 637. doi:10.3390/en10050637.

Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A., 2002. Collaborative conceptual design—state of the art and future trends. Computer-Aided Design 34, 981–996. doi:10.1016/S0010-4485(01)00157-9.

Welle, B., Haymaker, J., Rogers, Z., 2011. Thermalopt: A methodology for automated bim-based multidisciplinary thermal simulation for use in optimization environments. Building Simulation 4, 293–313. doi:10.1007/s12273-011-0052-5.

Wetter, M., Bonvini, M., Nouidui, T.S., 2016. Equation-based languages–a new paradigm for building energy modeling, simulation and optimization. Energy and Buildings 117, 290–300. doi:10.1016/j.enbuild.2015.10.017.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms — a comparative case study, in: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (Eds.), Parallel Problem Solving from Nature — PPSN V, Springer Berlin Heidelberg. pp. 292–301. doi:10.1007/BFb0056872.

Appendix A. Structural Design Grammar Settings

This appendix lists the used settings for the design grammars that are presented in this work. In table 1, the live load and the wind loads are given. Thereafter, in tables A.1-A.4 the structural properties of the components in the structural model are specified. Table A.1 specifies the flat shell properties, table A.2 the properties of beams, table A.3 the properties of trusses, and finally table A.4 gives the properties used for the substitute components.

Table A.1. The structural properties that apply to components of type `flat shell`.

Property type [−]	Thickness t in [mm]	Young's modulus E in $\left[\mathrm{N\,mm^{-2}}\right]$	Poisson's ratio ν [−]
1	150	30000	0.3

Table A.2. The structural properties that apply to components of type `beam`.

Property type [−]	Width w in [mm]	Height h in [mm]	Young's modulus E in $\left[\mathrm{N\,mm^{-2}}\right]$	Poisson's ratio ν [−]
1	150	150	30000	0.3

Table A.3. The structural properties that apply to components of type `truss`.

Property type [−]	Cross sectional surface A in $\left[\mathrm{mm^{2}}\right]$	Young's modulus E in $\left[\mathrm{N\,mm^{-2}}\right]$
1	22500	30000

Table A.4. The structural properties that apply to components of type `substitute`.

Property type [−]	Thickness t in [mm]	Young's modulus E in $\left[\mathrm{N\,mm^{-2}}\right]$	Poisson's ratio ν [−]
1	150	0.03	0.3

Table A.5. The structural loads that will be applied by a structural design grammar.

Type [−]	Load case [−]	Magnitude $\left[\mathrm{N\,mm^{-2}}\right]$	α_{az} [°]	α_{alt} [°]
live load	1	0.005	0	270
wind pressure	2	0.001	0	0
wind shear	2	0.0004	0	0
wind suction	2	0.0008	0	0
wind pressure	3	0.001	90	0
wind shear	3	0.0004	90	0
wind suction	3	0.0008	90	0
wind pressure	4	0.001	180	0
wind shear	4	0.0004	180	0
wind suction	4	0.0008	180	0
wind pressure	5	0.001	270	0
wind shear	5	0.0004	270	0
wind suction	5	0.0008	270	0

Appendix B. Building Spatial Designs

In this appendix, the details of the building spatial designs that are used in this work are presented. Figures B.1-B.3 show the designs that are used in the first case study of this work. Figure B.1 shows the design of a tall building with a central core. In figure B.2, a typical design of an apartment building with horizontal galleries is depicted. A large hall is shown in figure B.3, which is a common design for large industrial applications. Finally, figure B.4 presents the design of a portal shaped building which has been used for the second case study in this work.

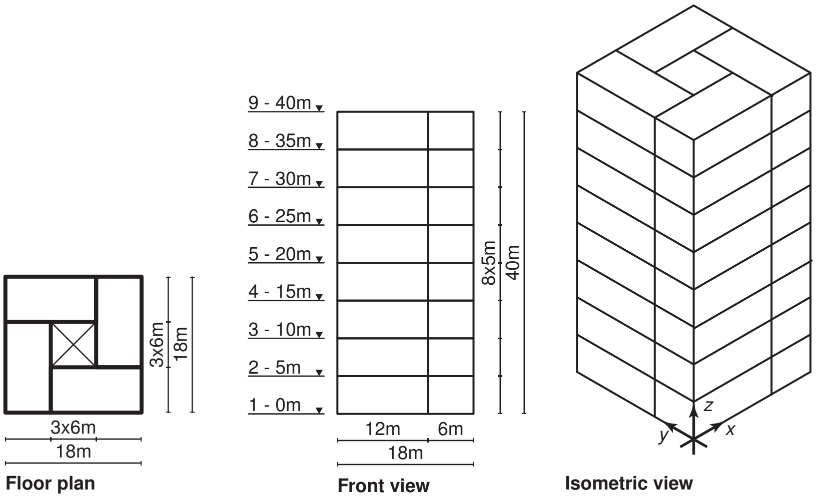

Fig. B.1. Design 1, a tall building.

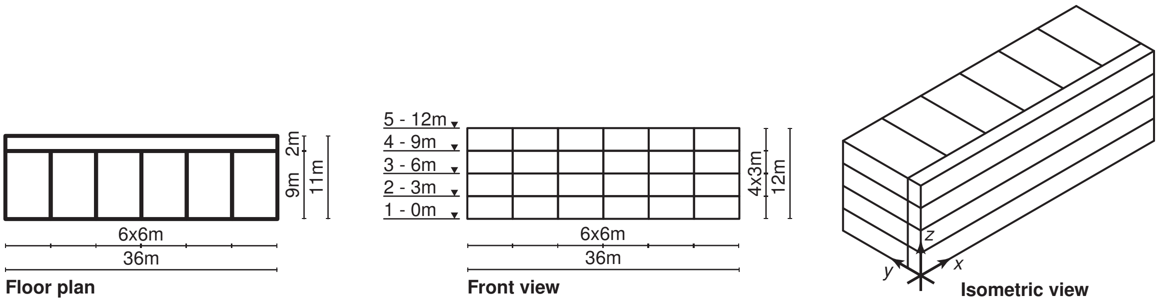

Fig. B.2. Design 2, an appartment building.

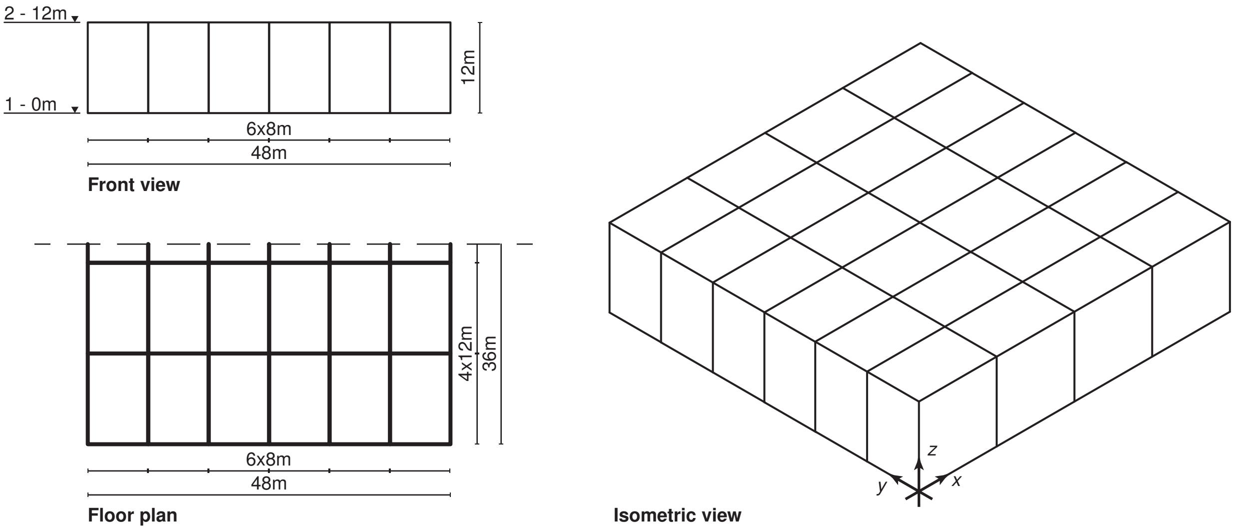

Fig. B.3. Design 3, a large hall.

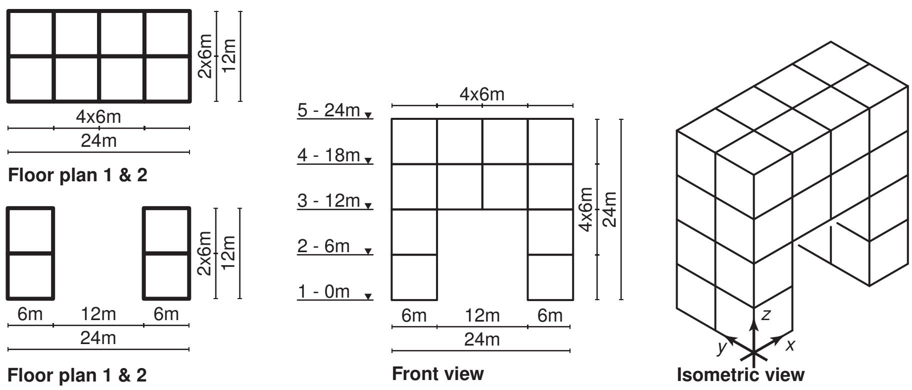

Fig. B.4. Design 4, a portal shaped building.