

# Parametrization-based solution space exploration for Model Healing

Jiabin Wu, Stavros Nousias, André Borrmann  
Chair of Computational Modeling and Simulation  
Technical University of Munich, Germany  
[j.wu@tum.de](mailto:j.wu@tum.de)

**Abstract.** Building Information Modelling (BIM) technologies have significantly improved building design efficiency. BIM-based Automatic Code Compliance Checking (ACCC) provides a robust way to evaluate whether building designs meet the construction codes and regulations. However, architects and designers usually have to alter the model manually to achieve code compliance by investigating the checking results. This process is effortful and error-prone since fulfilling one building rule might cause deficiencies in other requirements. To address this issue, this paper presents a solution space exploration workflow to find code-compliant building designs that are as close as possible to the original design. The proposed approach links the check results dynamically to design parameters. With sensitivity analysis, the initial high-dimensional solution space is dimensionally reduced. The identified valid alternatives constitute feasible regions where designs fulfill the regulations and are close to the original design. This paper contributes to automating the search for code-compliant designs close to the designers' intent, facilitating the model improvement process after compliance checking.

## 1. Introduction

Among the prominent methods that have significant potential for design improvement in Building Information Modeling (BIM) is the Automatic Code Compliance Checking (ACCC) (Eastman *et al.*, 2009). In ACCC, the proposed building design is checked for compliance with building codes and regulations without modifying the design. Available ACCC tools allow checking building designs for compliance with construction codes and regulations.

However, a significant challenge is that so far no automated solutions have been provided for fixing design errors (Lee *et al.*, 2019). Investigating the checking results, architects and designers tune the design parameters and adjust the model manually to achieve code compliance. The improvement process is laborious and error-prone, with the potential to “run in circles.” The rule-checking tools generate results entailing a list of violations identifying the violated rules and the involved building elements. However, the provided information is insufficient for fixing design errors. Although varying design parameters can help designers adjust the initial design, there are many parameters resulting in a high-dimensional solution space. Thus, it is computationally expensive to determine the optimal options in high-dimensional intervals of design parameters. Moreover, satisfaction with one checking rule might cause deficiencies in requirements from other aspects.

From a formal viewpoint, adjustments made by designers can be considered as searching for an optimum within a solution space formed by all possible design variations (Wu *et al.*, 2022). Applying optimization theory, valid (“feasible”) solutions are those that fulfill all requirements, and the objective function (representing the optimality criterion) represents the distance to the origin. i.e., the similarity with the design intended by the human.

This paper presents a solution space exploration workflow to address the aforementioned issues, providing feasible regions with designs fulfilling the checking requirements and being close to the initial design intent. We initialize the original solution space by low-discrepancy variations

of the initial design via quasi-random sampling. We employ sensitivity analysis to identify the most sensitive parameters regarding the checking rules. The space dimension is reduced by restricting unessential design parameters. Afterwards, the reduced solution space is explored deeper by increasing the relevant parameters' variation boundaries until achieving a sufficient number of compliant variants. We identify preliminary feasible regions within the reduced space based on valid clusters. Additional samples are explored to fill the preliminary feasible regions and the space between the initial design and the feasible regions. Finally, the valid and similar variants are determined to minimize deviation within feasible regions.

The parametrization-based solution space evolves from high-dimensional to reduced spaces with feasible regions meaningful for optimal search. This workflow links checking results dynamically to design adjustments, allowing effective design adjustment regarding code compliance checking. Direct benefits to architects and engineers can be gained from a broad extension and adoption of the workflow. This research also enhances the flexibility of automatic rule checking in achieving high design efficiency in the broad context of building design.

The rest of the paper is organized as follows: Section 2 presents state-of-the-art methods and related works. Section 3 describes the solution space exploration workflow. Section 4 is dedicated to the experimental setup and results, and Section 5 finally concludes the article.

## **2. Related works**

### **2.1 Code compliance checking**

Building codes are sets of rules laying down the minimum standards for building structures. Building design and construction rely on standardized checks to ensure safety in every aspect of the building. By executing standardized checks, designers avoid unnecessary design mistakes, making the design suitable for building permission.

ACCC assesses building designs in four stages: rule interpretation, building model preparation, rule execution, and rule check reporting (Eastman *et al.*, 2009). Assessing BIM models with digitalized regulatory requirements, ACCC frees designers from design checks in text formats to have more time to dedicate to complex and creative tasks. Available automated checking tools find violations between the BIM models and prescriptive building regulations. However, ACCC has limited guiding effects on finding appropriate design modification intent. With existing checking tools, human expert intervention is required to fix the detected design issues (Amor and Dimyadi, 2021). Designers interpret the failure reasons and adjust their designs with the pass or fail result generated from the kernels, which is typically an iterative step. Most design modifications are laborious and error-prone (Lee *et al.*, 2019; Wu *et al.*, 2022), with the potential to “run in circles” when designers manually search for appropriate alternatives.

ACCC eventually aims to improve the efficiency of the whole design and construction process. Amor and Dimyadi (2021) pointed out that one of the most promising potentials of ACCC is to support correct BIM model generation. By combining compliance checking, BIM object libraries, and design strategies, designs that meet regulatory requirements can be achieved. Incorporating prescribed regulatory requirements into the design is one method that leads to compliant designs (Dimyadi *et al.*, 2016). Prescriptive regulations typically specify the limit values of particular building elements, which can be embedded as prerequisite conditions. In this way, the relevant criteria are fulfilled. However, the architectural design of complex

buildings is typically a human creativity-driven process that has many compliance restrictions to incorporate for the elements and parameters.

On the other hand, correct models can be reached based on the identified rule violations. Lee *et al.* (2019) developed a design support system to provide design recommendations after identifying the code compliance on fire safety, contributing to the correct model generation. The proposed design support system detects the alternatives and gives recommendations with a ranking list. However, the recommendations focus on ontology and semantic issues without dealing with space and geometry-related design problems, which typically require repeated rectification. Investigating locational and geometric factors is essential in reducing the risk of non-compliance and its associated improvement costs in building design.

The concept of “healing” a design model aims at correcting a given BIM to fulfill requirements and constraints (Collins *et al.*, 2022; Forth *et al.*, 2022). One application is to assign correct element types and materials to the respective model elements (Forth *et al.*, 2022). Another application addressed here is to resolve violations of codes by altering the initial model (Wu *et al.*, 2022). The model adaptation framework selects design alternatives that comply with the building regulations and deviate the least from the initial design. The proposed approach tackles the issue of manual model adjustment by gathering comparable design variants and measuring their dissimilarity to the origin. However, the complexity in the considered design conformance is limited to the element level, hampering its broad applicability. Besides, overcoming non-compliance is usually a multicriterial process in case there are many checking aspects. Especially for space and geometry-related factors, fulfilling one checking aspect might cause deficiencies in other correlative aspects.

## 2.2 Solution space

Solution space can analytically represent the potential solutions in a simplified way for engineering and design problems defined by variables and objectives. It can be described as a multidimensional space from the product of variables’ intervals (Markus and Johannes, 2012). The BIM technology allows building designs to be created parametrically (Sacks *et al.*, 2004), providing design control through altering parameters. Accordingly, each one of the related design parameters can be represented by one dimension of the solution space.

Wu *et al.* (2022) varied preselected parameters to form a solution space for Model Healing. In the context of Model Healing, the design problem is to fix the detected design issues, and the objective function represents the “distance” from adapted designs to the original design. Particular combinations of parameters result in valid solutions, while others will not, thus outlining the potential subspaces for feasible and infeasible solutions. Feasible solutions refer to potential combinations of values for the variables that meet the given requirements, each of which refers to a point in the solution space (Kang *et al.*, 2011; Lee *et al.*, 2018; Wu *et al.*, 2022). The design adaptation starts from a solution in the infeasible region and searches for feasible solutions close to the original design. However, unsuitable combinations may alter the initial design topology or cause a clash between building elements. Thus, fixing the variation boundaries can ease the variant sampling process within the solution space. Moreover, the solution space variables should be decoupled to keep independence among all the dimensions to ensure sampling efficiency (Graff *et al.*, 2016).

For specific problems, e.g., code-compliant design searching, the feasible solutions are typically located within one or multiple regions of the parameter space. The solution search process may involve too many points to determine the optimal solutions, leading to high

computational costs. Thus, the range of parameter sampling is critical to define the extent of the solution space (Ortiz *et al.*, 2021). Rather than relying on one dense sample of the entire space, the solution space exploration should be refined iteratively by switching to the correct varying direction and zooming into promising regions (Markus and Johannes, 2012). In other words, the solution spaces should evolve shape and dimensionality throughout the exploration. Besides, the exploration process must consist formal representation of designs, machine-assisted evaluation techniques, and methods navigating to distinctive solutions (Kang *et al.*, 2011). Hu *et al.* (1999) developed a partitioning technique for designers to select space portions to explore after collapsing a multidimensional design space into a lower-dimensional space.

### 2.3 Sensitivity Analysis

Sensitivity analysis (SA) describes the derivative of the output values concerning model input variables. SA methods are often categorized as either local sensitivity analysis (LSA) or global sensitivity analysis (GSA). LSA relies on the one-at-a-time method and evaluates the significance of input parameters at specific points, neglecting effects from correlated input parameters or nonlinear behaviors. Distinct from LSA, GSA is a generic description of methods that estimates the effect of an input parameter on the output by varying all the parameters chosen (Kristensen and Petersen, 2016).

Pang *et al.* (2020) conducted a literature review on SA methods in building performance analyses. Guo *et al.* (2022) tested multiple SA methods, including Morris and Sobol, to help architectural designers identify the most efficient energy-saving parameter strategy. SA can determine the essential design variables, facilitating the design process for specific problems.

Selecting suitable SA methods depends on defined problems (Borgonovo and Plischke, 2016; Pang *et al.*, 2020). The variance-based SA methods, such as the FAST (Mc *et al.*, 1982) and Sobol methods (Sobol, 2001), can obtain stable results for non-monotonic and nonlinear problems (Pang *et al.*, 2020). FAST calculates the main effect of individual variables on output variability without considering the variables' interactions. The Sobol sequence comprehensively explores input variables' effects on output results (Sobol, 2001; Saltelli *et al.*, 2010; Owen, 2022). The Sobol method assesses the sensitivity by evaluating single and multiple variables' contribution to the output variability but tends to require many model evaluations.

### 3. Methodology

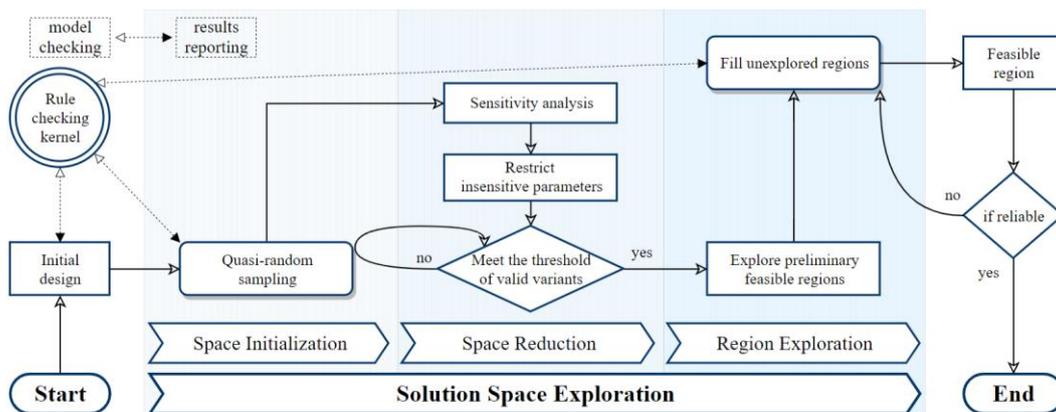


Figure 1 Solution space exploration workflow for Model Healing

We propose the following 3-step solution space exploration workflow (Figure 1) to achieve Model Healing. The workflow investigates computational sampling methods and characterizes parametric building designs in evolving solution spaces.

### 3.1 Space initialization

The initial design is primarily characterized by design parameters  $x = (x_1, \dots, x_n)$ . Based on the compliance checking results, we define a design evaluation criterion  $y$ :

$$y = v - v_{req} \quad (3.1)$$

with  $v$  being the actual value of the conformance checked, and  $v_{req}$  the corresponding value required in building codes. We initialize the solution space with low-discrepancy variants by quasi-random sampling based on the Sobol sequence (Sobol, 2001). Each sampled parameter  $\bar{x}_i$  is perturbed with uniform random noise so that  $\bar{x}_i = x_i + U(-l, l)$  where  $l$  is an arbitrary value defined experimentally. Without the risk of violating design constraints, the sampling is done with relatively small intervals for all the related design parameters, constituting a vast multidimensional solution space centered around the initial design. Every variant is evaluated with the compliance checker when created, labeling the solution as a compliant or non-compliant option with a scalar output  $y$ .

### 3.2 Space reduction

Based on the quasi-random sampled Sobol sequence, we employ sensitivity analysis to decompose the output variability into design parameters. The relation between the evaluation criterion and the design parameters is assumed as a non-formalizable and integrable nonlinear function  $f$ :

$$y = f(x) = f_0 + \sum_i f_i(x_i) + \sum_{i < j} f_{ij}(x_i, x_j) + \dots + f_{12\dots n}(x_1, x_2, \dots, x_n) \quad (3.2)$$

With  $1 \leq i_1 < \dots < i_s \leq n$ , the constant part  $f_0$ , and other high-order parts: e.g.,  $f_i$  only depends on  $x_i$ , and  $f_{ij}$  is associated with  $(x_i, x_j)$ . Assuming all parameters are independently varied and  $f(x)$  is square integrable, the variance  $V$  and the partial variance  $V_{i_1 \dots i_s}$  are defined:

$$V = \int f^2(x) dx - f_0^2, \quad V_{i_1 \dots i_s} = \int f_{i_1 \dots i_s}^2 dx_{i_1} \dots dx_{i_s}, \quad V = \sum_{s=1}^n \sum_{i_1 < \dots < i_s} V_{i_1 \dots i_s} \quad (3.3)$$

To estimate the influence of design parameters on design compliance, the global sensitivity indices  $S_{i_1 \dots i_s}$  are calculated, where  $s$  is the order of the index. For example,  $S_{i_1}$  represents the first-order effect of  $x_{i_1}$ ,  $S_{i_1 i_2}$  expresses the second-order effect of  $(x_{i_1}, x_{i_2})$ . The total-order index  $S_{i_1}^T$  measures the contribution of  $x_{i_1}$  to the output caused by all its interactions.

$$S_{i_1 \dots i_s} = \frac{V_{i_1 \dots i_s}}{V}, \quad S_{i_1}^T = S_{i_1} + S_{i_1 i_1} + S_{i_1 i_2} + \dots + S_{1 \dots i_1 \dots (n-1)n} \quad (3.4)$$

Via a Monte Carlo algorithm, global sensitivity indices are estimated by using  $y$  values only (Sobol, 2001; Saltelli *et al.*, 2010; Owen, 2022). We determine the essential parameters  $x^* \in x$  and separate out the insensitive parameters to reduce the solution space's dimension.

Meanwhile, the sampling boundaries for essential parameters  $x^*$  is increased gradually until achieving a required amount of valid design alternatives.

### 3.3 Region exploration

Preliminary feasible regions are clustered within the reduced solution space according to groups of valid alternatives. We utilize K-means clustering to partition the samples into multiple clusters based on the relevant parameters' mean values. Since the space has been dimensionally reduced to a lower-dimensional subspace, the simplest random initialization is employed.

Based on the preliminary region that contains the most valid alternatives, we refine the following unexplored regions by random sampling: 1) the identified regions and 2) the routes from the original design to the preliminary regions by varying the essential parameters  $x^*$ . Filling those two partially-explored regions ensures the reliability of the identified cluster, and the feasible region is the closest one to the initial design.

## 4. Proof of concept

### 4.1 Scenario description

A case study concerning building rules of the 2018 International Building Code (IBC 2018) (International Code Council (ICC), 2021) is provided to demonstrate the applicability of the workflow. The main descriptions of the selected rules are listed in Table 1. The rule-checking kernel is implemented in Dynamo for Autodesk Revit and executed on a one-storey building model (Figure 2). In this study, we focus on geometry and space-related building rules.

Table 1: The analyzed building rules in IBC 2018 (International Code Council (ICC), 2021)

Subject	Outlined Description
1020.2: Width and capacity	The required capacity of corridors shall ..., but the minimum width shall ...
1207.1: Minimum room widths	Habitable spaces, other than ..., shall be ... in any plan dimension.
1207.3: Room area	Every ... shall have not less than ... of net floor area. Other ... shall ...

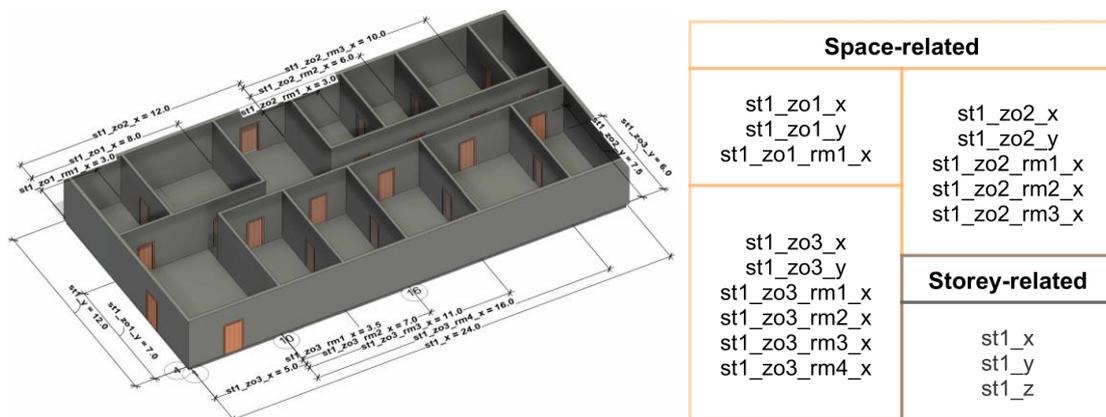


Figure 2: One-storey building model (left) and its geometry- and space-related parameters (right)

## 4.2 Results

The variation ranges of all space-related parameters are initially set to  $l = 0.02\text{ m}$  for sensitivity analysis. The ranges gradually expand to enlarge the space exploration scope. The valid variant threshold amount is set to 10. We summarize the sampling results in Table 2.

Table 2: Sampling details and results for solution space exploration

No.	Exploration objectives	Sampling methods	Variation ranges	Number of		
				Varied parameters	Variants	Valid variants
1	Initialize, reduce dimensions	Quasi-random sampling (Herman and Usher, 2017)	$[-0.02\text{ m}, 0.02\text{ m}]$	14	1920	0
2	Enlarge the reduced space		$[-0.1\text{ m}, 0.1\text{ m}]$	4	320	0 (<10)
3	Enlarge the reduced space		$[-0.5\text{ m}, 0.5\text{ m}]$	4	160	14 (>10)
4	Fill unexplored regions		1) identified regions; 2) routes from the origin	4	910	186

Figure 3 illustrates the sensitivity analysis outcomes for each related rule in the phase of space reduction. With the first- and total-order indices, the four most important parameters are selected to enlarge the scope of the dimensionally reduced space and search valid variants.

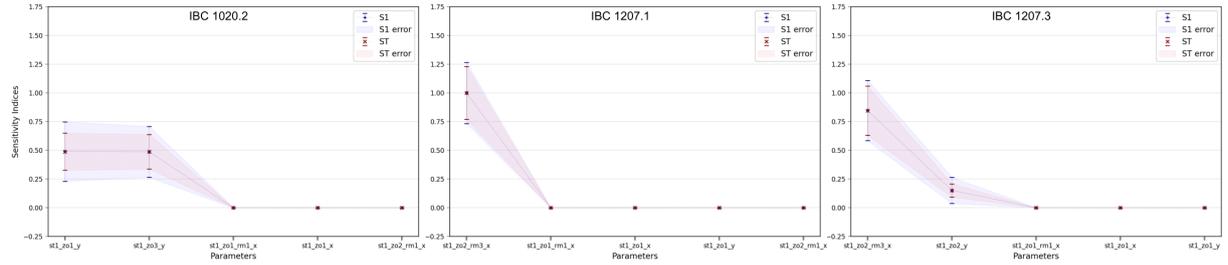


Figure 3: First- and total-order sensitivity indices of (the five most essential) parameters

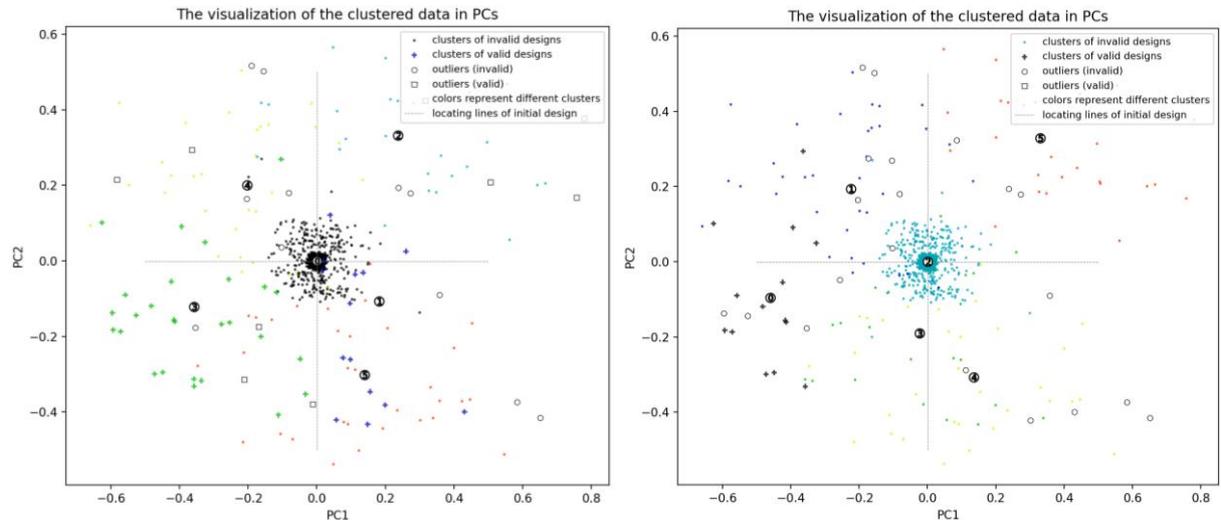


Figure 4: Identified preliminary feasible regions for IBC1020.2 (left: clusters 3 and 5) and all selected IBC rules (right: cluster 0) via K-means clustering

Preliminary feasible regions have been identified by applying K-means clustering, illustrated with the first two principal components in Figure 4. The identified feasible regions represent sub-spaces that include all the variants observed as compliant. Two preliminary feasible regions are identified for the rule IBC1020.2 and one for the three selected IBC rules. Besides, we calculate outliers dissimilar to the rest of the samples in each cluster based on the difference in related parameters. We remove those anomalous samples for each cluster, especially those for valid designs, to make the preliminary feasible region more reliable.

Refining the preliminary region and exploring the routes from the origin to the preliminary regions (Figure 5), the feasible region is updated with more detected valid variants. Meanwhile, no other potential valid variants closer to the origin have been detected, ensuring the identified cluster is the closest feasible region to the initial design (against all three rules). The samples with negative silhouette coefficients within cluster 1 are close to the (in)feasibility boundaries.

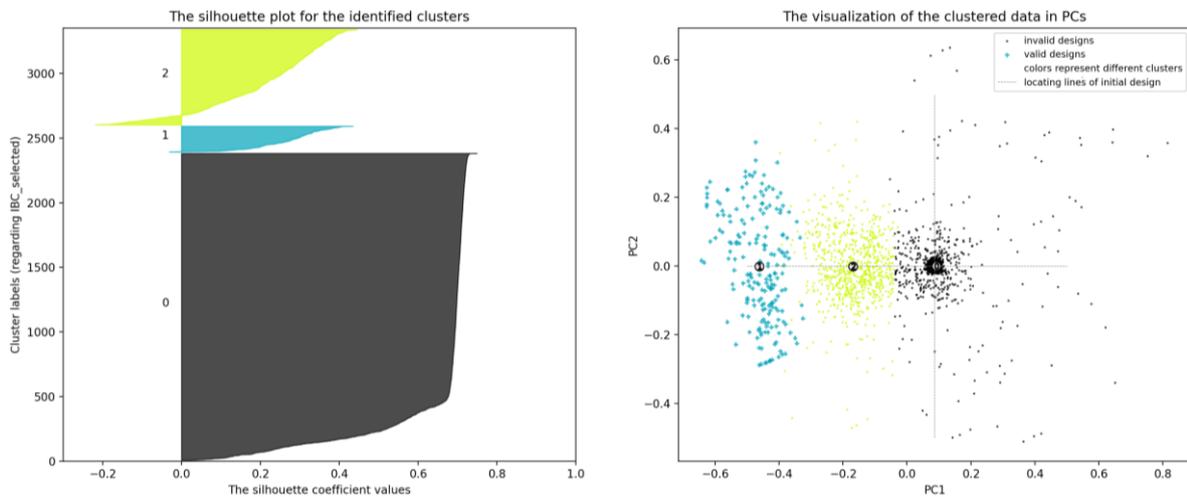


Figure 5: Verification of the preliminary feasible region: the silhouette values for the clusters labeled by K-means cluster analysis (left), and cluster 1 is the updated feasible region (right)

After the feasible region is verified, the design variants that fulfill the requirements are identified. We project all the samples back to the solution space in a coordinate system constituted by actual design parameters. Figure 6 introduces invalid variants outside the identified feasible region and code-compliant variants in the feasible region.

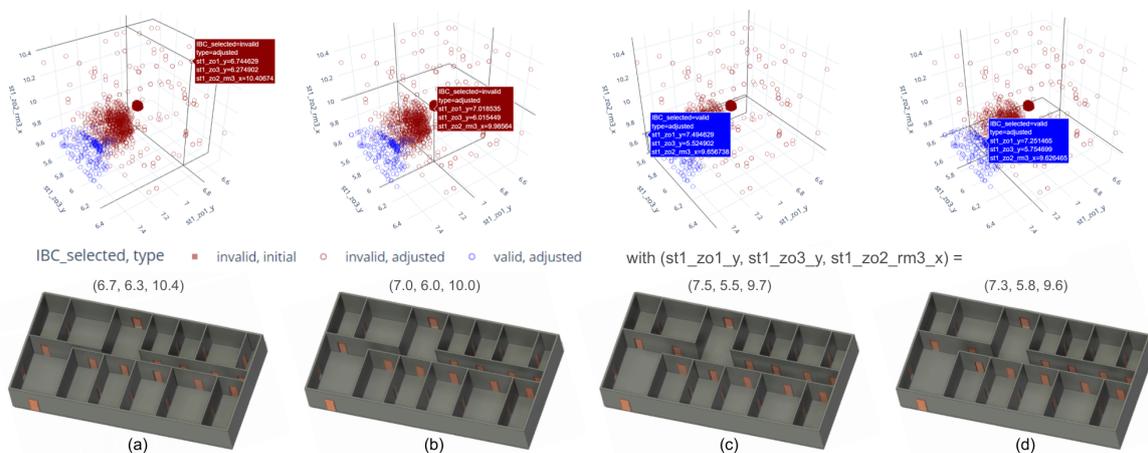


Figure 6: Design variants outside (a, b) and situated within (c, d) the feasible region. The latters represent the healed models

## 5. Discussion & Conclusion

The proposed approach provides solutions for designs that violate building codes and regulations. Using the notions of optimization theory, we set the objective function by the distance to the original design within the high-dimensional solutions space, in which each parameter corresponds to a coordinate axis. Since feasible region searching in high dimensional spaces is computationally expensive, we perform a sensitivity analysis to reduce the dimensions to the most relevant ones. We determine the feasible regions in the refined solution space to locate code-compliant alternatives, guiding the design improvement efficiently.

This workflow's objective differs from design support systems determining the best solutions for an optimization problem, where the objective function reflects the building's performance, e.g., in terms of carbon footprint. The solution space exploration for Model Healing investigates the non-compliance issues against multiple building rules. It determines reliable regions for efficiently finding compliant alternatives in further design steps. Despite the similarity with SA in building performance optimization, this workflow focuses on model correction for the detected geometry and space-related issues from code compliance checking.

In this study, we investigate code-compliant solutions when maintaining the design topology, while the design adjustment process may involve more complex variations, e.g., altering the relationship between model elements. The proposed approach can be ameliorated by adding semantics evaluation in larger-scale building models. This would entail dynamic sampling strategies that process non-numeric values and consider the intervention of parameter dependencies. A comprehensive BIM parametrization will ease the adjustment process to cope with various objectives of rule-checking. Other future challenges would be to compute the parameter variation ranges and represent the design constraints in the given model.

## 6. Acknowledgments

This research was funded by the TUM International Graduate School of Science and Engineering (IGSSE).

## References

- Amor, R. and Dimyadi, J. (2021) 'The promise of automated compliance checking', *Developments in the Built Environment*, 5(October 2020), p. 100039. doi: 10.1016/j.dibe.2020.100039.
- Borgonovo, E. and Plischke, E. (2016) 'Sensitivity analysis: A review of recent advances', *European Journal of Operational Research*, 248, pp. 869–887. doi: 10.1016/j.ejor.2015.06.032.
- Collins, F. C. et al. (2022) 'Shape encoding for semantic healing of design models and knowledge transfer to Scan-to-BIM', *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, 175, pp. 1–21. doi: 10.1680/jsmic.21.00032.
- Dimyadi, J., Amor, R. and Spearpoint, M. (2016) 'Using BIM to support simulation of compliant building evacuation', *eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016*, (September), pp. 511–520.

- Eastman, C. et al. (2009) ‘Automatic rule-based checking of building designs’, *Automation in Construction*, 18(8), pp. 1011–1033. doi: 10.1016/j.autcon.2009.07.002.
- Forth, K., Abualdenien, J. and Borrmann, A. (2022) ‘NLP-based Semantic model healing for calculating embodied carbon in early building design stages’, in *Proc. of European Conference on Product and Process Modeling*.
- Graff, L., Harbrecht, H. and Zimmermann, M. (2016) ‘On the computation of solution spaces in high dimensions’, *Structural and Multidisciplinary Optimization*, 54(4), pp. 811–829. doi: 10.1007/s00158-016-1454-x.
- Guo, J. et al. (2022) ‘Energy Prediction and Optimization Based on Sequential Global Sensitivity Analysis: The Case Study of Courtyard-Style Dwellings in Cold Regions of China’, *Buildings*, 12(8), p. 1132. doi: 10.3390/buildings12081132.
- Herman, J. and Usher, W. (2017) ‘SALib: An open-source Python library for Sensitivity Analysis’, *Journal of Open Source Software*, 2(9), p. 97. doi: 10.1016/S0010-1.
- Hu, X. et al. (1999) ‘Framework for user assisted design space exploration’, *Proceedings - Design Automation Conference*, pp. 414–419. doi: 10.1145/309847.309970.
- International Code Council (ICC) (2021) *International Building Code 2018*. Available at: <https://codes.iccsafe.org/content/IBC2018P6> (Accessed: 28 November 2022).
- Kang, E., Jackson, E. and Schulte, W. (2011) ‘An approach for effective design space exploration’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6662 LNCS, pp. 33–54. doi: 10.1007/978-3-642-21292-5\_3.
- Kristensen, M. H. and Petersen, S. (2016) ‘Choosing the appropriate sensitivity analysis method for building energy model-based investigations’, *Energy and Buildings*, 130(October), pp. 166–176. doi: 10.1016/j.enbuild.2016.08.038.
- Lee, D. et al. (2018) ‘Design space exploration of 3D network-on-chip: A Sensitivity-based Optimization Approach’, *ACM Journal on Emerging Technologies in Computing Systems*, 14(3). doi: 10.1145/3197567.
- Lee, P. et al. (2019) ‘An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning’, *Construction Management*, 23, pp. 1952–1962. doi: 10.1007/s12205-019-1750-2.
- Markus, Z. and Johannes, E. von H. (2012) ‘Computing solution spaces for robust design’, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, (February), pp. 1102–1119. doi: 10.1002/nme.
- Mc, G. J. et al. (1982) ‘Global sensitivity analysis—a computational implementation of the Fourier amplitude sensitivity test (FAST)’, *Computers & Chemical Engineering*.
- Ortiz, J. et al. (2021) ‘A Topological Formalism for quantitative analysis of design spaces’, *Proceedings of the Design Society*, 1(August), pp. 293–302. doi: 10.1017/pds.2021.30.
- Owen, A. B. (2022) ‘On Dropping the First Sobol’ Point’, *Springer Proceedings in Mathematics and Statistics*, 387(Mc), pp. 71–86. doi: 10.1007/978-3-030-98319-2\_4.
- Pang, Z. et al. (2020) ‘The role of sensitivity analysis in the building performance analysis: A critical review’, *Energy and Buildings*. Elsevier, p. 109659. doi: 10.1016/j.enbuild.2019.109659.
- Sacks, R., Eastman, C. M. and Lee, G. (2004) ‘Parametric 3D modeling in building construction with examples from precast concrete’, *Automation in Construction*, 13(3), pp. 291–312. doi: 10.1016/S0926-5805(03)00043-8.
- Saltelli, A. et al. (2010) ‘Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index’, *Computer Physics Communications*, 181(2), pp. 259–270. doi: 10.1016/j.cpc.2009.09.018.
- Sobol, I. M. (2001) ‘Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates’, *Mathematics and Computers in Simulation*, 55, pp. 271–280.
- Wu, J. et al. (2022) ‘Model Healing: Toward a framework for building designs to achieve code compliance’, in *Proc. of European Conference on Product and Process Modeling*. CRC Press, pp. 450–457. doi: 10.1201/9781003354222-58.