Embedding Design Intent into Performance-Based Architectural Design—Case Study of Applying Soft Constraints to Design Optimization

DongLai Yang, Likai Wang(✉), and Ji Guohua

Nanjing University, Hankou Road 22, Nanjing, Jiangsu, China
wang.likai@nju.edu.cn

Abstract. The lack of consideration of subjective design intents hinders the application of performance-based design optimization to architectural design because building performance is not the only aspect that designers need to solve. In response, this study proposes a method integrating subjective design intents into performance-based design optimization using soft constraints. To demonstrate the method, a case study is presented, where the design optimization continuously provides feedback to the designer and helps them reformulate and redefine the design problem. The case study shows how the application of design optimization and soft constraints is able to assist designers in identifying implicit and hidden design problems and stimulate design exploration at the early design stage.

Keywords: Performance-based design · Design intent · Soft constraints · Optimization · Co-evolution

1 Introduction

Performance-based design optimization, which integrates parametric models, building performance simulations, and evolutionary optimization, has been widely considered an effective design tool for sustainable architectural design. Many studies have demonstrated its role in addressing complex performance challenges in building design. However, other factors in architectural design, such as functionality and aesthetics, are often omitted in research. This tendency is also reinforced by the notion that design intents are difficult to quantify [5]. The claim greatly affects the application of performance-based design optimization to real-world architectural design tasks. As a result, the design optimization is often conducted after the design scheme is determined, thereby, separating it from the conceptual development process.

In architectural design, designers have to integrate various factors into the design synthesis process, including functionality and aesthetics that are judged subjectively and building performance factors that are evaluated objectively. In order to incorporate subjective intents into the optimization, several approaches have been explored, such as using interactive genitive algorithms (IGA) [2, 3] or aesthetic-related constraints [9].

P. F. Yuan et al. (eds.), *Hybrid Intelligence*, Computational Design and Robotic Fabrication,
https://doi.org/10.1007/978-981-19-8637-6_14

The use of these approaches enables architects to intervene in the optimization process and allow architects' personal preferences to be integrated into the process of design optimization, but they are not without problems. First, using these methods can still result in a huge number of unfeasible designs generated if using under-constrained or naive generative models. Second, interactive approaches, such as IGAs, require architects to spend considerable time and energy to select or score the generated designs, which can disrupt architects' design processes. Last but not least, the feedback loop between architects' design development and performance-based design optimization is also absent from most existing studies.

1.1 Paper Overview

Considering the limitation of the previous studies, this study proposes an approach to integrating subjective design intents into performance-based design optimization using soft constraints. With the use of soft constraints, specific design intents, such as view, building forms, and site constraints, can be formulated into the fitness evaluation by using penalty or award functions [1]. As a result, designers are enabled to navigate the optimization search and make the optimization produce more desieable designs that can both satisfy the performance objective and design intent.

To demonstrate the efficacy of the proposed approach, a case study is presented in the paper, where the design is started only considering performance factors. Then, through reflecting on the optimization result, we iteratively insert factors related to functionality into the fitness evaluation and make the optimization result achieve an acceptable compromise between the performance improvement and design intent. The case study shows that the proposed design approach can strengthen the feedback loop between designers and computers, making the designer more engaged in the design development process informed and inspired by performance-based design optimization. This design process can be viewed as a "meta-optimization" process where the objective is not merely focused on performance improvement but also to achieve a "co-evolution" between designers and computers to attain a well-rounded design.

2 Method

Early-stage architectural design is widely accepted as an iterative design exploration process. Therefore, it is also critical for the computational design tools or design approaches to support the iterative and *human-in-the-loop* design process. In light of this, the proposed design optimization is envisioned as continuously providing feedback to the designer from the outset of the design process rather than offering specific and determined solutions (Fig. 1). In other words, it encourages architects to reflect on the optimization result and iteratively reformulate the design objective with the use of computational design optimization.

The workflow is built on the Rhino-Grasshopper platform. EvoMass and other building performance simulation tools, such as Ladybug and ClimateStudio, are used in the

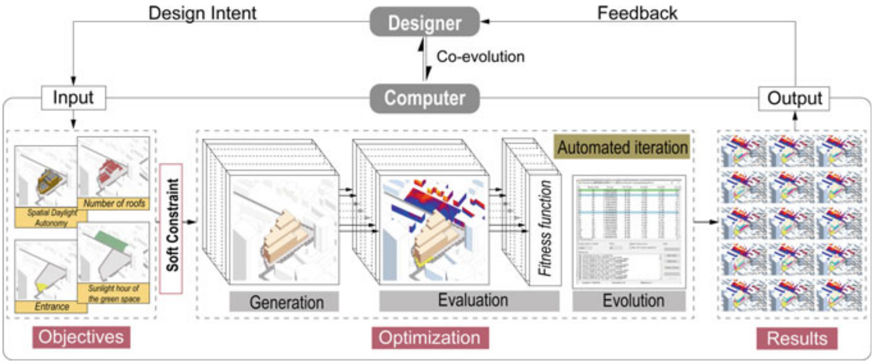

Fig. 1. Proposed optimization-based design workflow

design optimization workflow. The combination of these tools has already been applied to performance-oriented design optimization. However, previous applications fell short of the pursuit of satisfying subjective design intents. Therefore, to make architects' design intents to be included in the optimization process, we introduce the application of soft constraints, which can effectively embed design intent into the optimization.

2.1 Design Generation and Optimization

In the proposed design optimization workflow. EvoMass serves as the building design generator and the optimization solver [8]. When using EvoMass for building design optimization, the designer first customizes the generative component in EvoMass to adapt the generated building massing design to the building site. There are two generative components in EvoMass built on the additive and subtractive form generation principles, and both components can generate diverse building massing designs, which can facilitate the optimization process to identify site- and task-specific solutions for various design projects.

Second, the generated building massing design is assessed by different design evaluation functions. For performance-based design optimization, simulation tools, such as Ladybug and ClimateStudio, are often used to measure the performance of the generated design. The design evaluation function will guide the optimization search direction. Therefore, in addition to performance factors, other design factors can be also included in the design evaluation function and, thereby, steer the search direction.

Third, the performance and the evolution of the design are converted into a fitness score and sent back to the optimization algorithm. When using EvoMass, the embedded evolutionary algorithm—SSIEA (Stead-State Island Evolutionary Algorithm), is used as the optimization solver to evolve the design population and identify the high-fitness solutions [8].

EvoMass can produce optimization results with diverse solutions that best satisfy the optimization objective. Furthermore, designers can re-evaluate the optimization result and modify the optimization objective. Previous applications of EvoMass mostly focused on building performance, while the capability of generating diverse building massing forms makes EvoMass an ideal form-finding tool for combining performance-based design and architectural design. Hence, this study further explores the potential of EvoMass in architectural design and investigates how the design optimization workflow can be intertwined with architects' design loop for conceptual development.

2.2 Soft Constraint

In evolutionary computing, constraint handling, including direct and indirect constraints, plays a critical role in solving optimization problems [1]. For direct constraints, the constraint is embedded into the design generation stage instead of in the design evaluation stage, using methods such as repair functions. This approach can effectively prevent invalid and chaotic solutions from being generated, while it often reduces the variability of the design generation, and it is possible to exclude promising solutions from the design search space [7].

For indirect constraints, the constraint is embedded into the design evaluation stage, using methods such as penalty or award functions. Regarding design applications, we further divide indirect constraints into hard and soft constraints. For hard constraints, designs that cannot meet the constraint will be directly eliminated and "killed" from the design population, which can rapidly narrow down the search scope and speed up the convergence of the optimization process. However, when using hard constraints, the population diversity will drop rapidly, and promising designs that even slightly violate the constraint will also be removed from the pool of recombination. It is because, for evolutionary optimization, the design optimization process heavily relies on the re-combination, namely crossover, of the genotype from different designs (typically two designs). Thus, to fully explore the design space, the evolutionary process needs to maintain an adequate population diversity that allows for the recombination of the design with heterogeneous genotypes.

In comparison, the application of soft constraints has the advantage of maintaining the population diversity during the optimization process. When using soft constraints, the fitness of the design that violates the constraint will be proportionally decreased to reduce its chance of surviving in the subsequent evolutionary process. Since the design remains in the design population, it can still be recombined with other designs. More importantly, if this design contains key parameters (genomes) that are essential parts of the genotype of high-fitness designs, the recombination with other designs may produce the offspring design that does not violate the constraint while having an advantageous fitness.

The application of soft constraints provides a feasible way for the designer to navigate the optimization process by converting the design intent into the design optimization process. Thus, in the following case study, we demonstrate how different design intents can be integrated into the optimization process using soft constraints and how the design optimization result shows the response to the design intent.

3 Case Study

In this study, we present a case-study design consisting of three stages and assume that the designer begins with the use of design optimization only focusing on performance factors and then iteratively integrates the design intent into the optimization process, including responding to the surrounding environment, functionality, and aesthetics. This case study describes an office building design in Nanjing, China. The building is located in the city center and is imposed a 50-m height limit (Fig. 2). There are several residential buildings on its west and north sides, urban green space on its northwest side, a main road on the southern side, and several high-rise office buildings on its east and west sides.

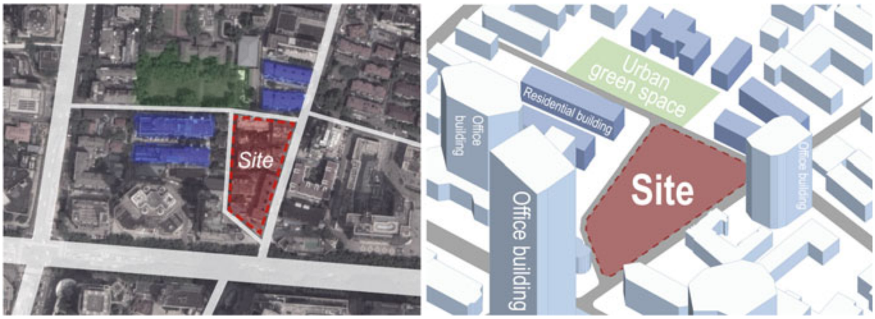

Fig. 2. Site overview

Within such a complex urban environment, only considering the building performance is insufficient. The irregular site boundary and the high-density urban environment pose a great challenge to the designer. In addition, as widely accepted as a "wicked" problem, architectural design often faces many hidden constraints that are not explicit to be identified at the outset of the design. As Schön [4] stated, conceptual design is a "*moving-seeing-moving*" process, where designers often discover implicit problems or constraints when manipulating the design object. As such, architects can leverage design optimization as an approach to uncovering hidden design problems and reformulate the design objective by superimposing the information gathered from design optimization.

In terms of building performance, daylight factor, spatial daylight autonomy, and discomfort glare have been commonly used in design evaluation. In this case study, the surrounding high-rise buildings cast a large shadow that can affect the daylighting quality of the target building. Thus, the spatial daylight autonomy (sDA) is first taken as the evaluation metric, simulated by ClimateStudio in Rhino-Grasshopper. Additionally, two soft constraints are applied to control the gross floor area (GFA) and density of the design.

According to the above-mentioned objectives, the initial fitness function of the first stage is shown in Fig. 3, where *p_area* and *p_den* represent the penalty function for GFA and density. For GFA, it calculates the difference between the actual GFA of each generated design and the target GFA (40,000 m^2) and proportionally decreases the fitness value according to the GFA difference. For the density, it punishes the design with a density outside the range of 0.6 to 0.8 and also proportionally decreases the fitness value based on the difference between the actual density and the target density.

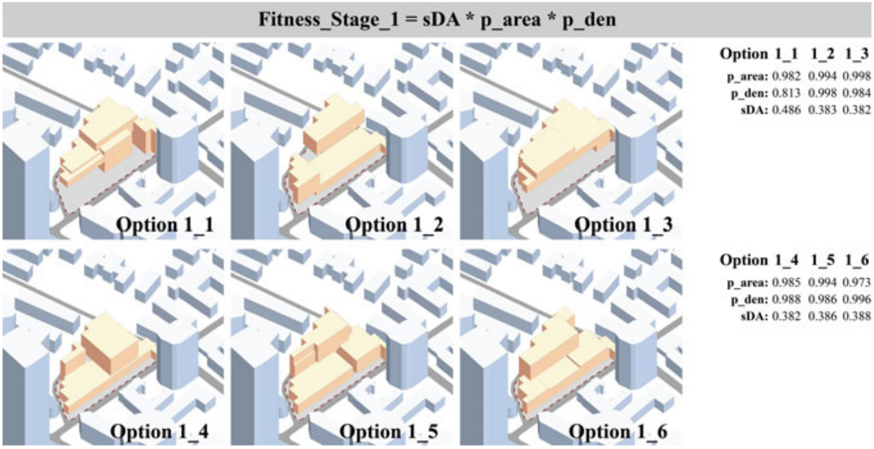

Fig. 3. The fitness evaluation function (top) and the optimization result in the first stage (bottom)

The above fitness function shows that when using soft constraints, there can be multiple objectives that need to be optimized. To handle multiple optimization objectives, the conventional approach based on Pareto optimization becomes inefficient as the goal of seeking as many trade-off (non-dominated) designs as possible can hinder the optimization progress. In addition, using Pareto optimization often results in too many design options, making it difficult to analyze and extract design information. In this regard, when using soft constraints, a more advisable approach is to use weight-sum and -product approaches. In this case study, we adopt a weight-product approach to integrate different optimization objectives. In comparison to weight-sum approaches, weight-product approaches do not require normalization as the change in each value can equally affect the overall fitness.

Figure 3 (bottom) shows the optimal design options found by the optimization process. It is not difficult to notice that these design options tend to keep away from the high-rise buildings in the south to escape from the shadow cast by these buildings. However, as the building massing typically gathers to the north to enhance its daylighting accessibility, this tendency significantly undermines the sunlight and daylight accessibility for other buildings and the surrounding public space

In the second stage, to decrease the adverse effects on the surrounding environment, the sunlight hour of surrounding residential buildings and the urban green space are included in the optimization. Sunlight hours are simulated using Ladybug in Rhino-Grasshopper, and the simulation only calculates the sunlight hour during winters. Therefore, by integrating the factors considered in the first stage, the fitness function of the second stage is shown in Fig. 4, where *sunlight_rsd* and *sunligh_green* respectively indicate the sunlight hour of the residential buildings and the green space. As a result, with the inclusion of the two new soft constraints, the fitness of the design is decreased according to the amount of sunlight blocked by the generated building massing form.

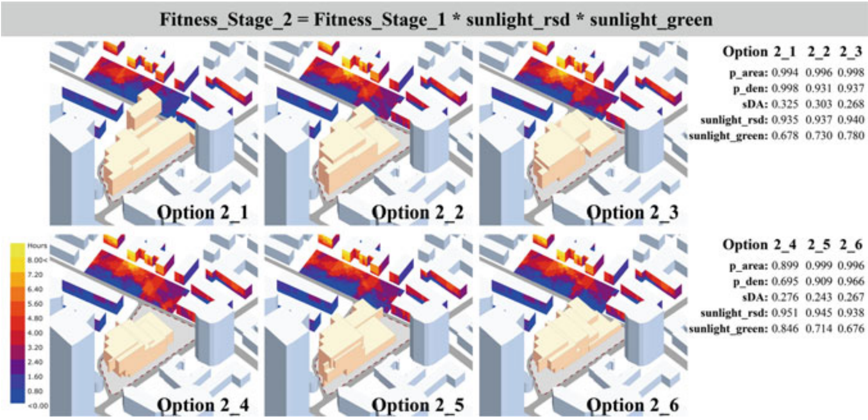

Fig. 4. The fitness evaluation function (top) and the optimization result in the second stage (bottom)

As shown in Fig. 4 (bottom), to reduce sunlight obstruction, the optimal design options typically have the massing volume retreating from the north, which can reduce the shadow cast on the residential building and the green space. However, to meet the GFA requirement, most of the massing volume is stacked and accumulated in the south, resulting in the building being too close to the surrounding office buildings. From an urban design perspective, this tends to make the outdoor space in the south of the target building over-crowded, which may also reduce urban ventilation.

In the third stage, four additional design intents are included. First, we place the entrance on the southern side of the target building to reduce the crowdedness in this area. Hence, we define a volume for the entrance space and punish the design based on how much the entrance space is "invaded" by the building's massing volume. Second, to enhance people's well-being, the percentage of the unobstructed view is also included in the optimization. Third, to allow for more outdoor and semi-outdoor spaces for the people working in this building, we award the design with more roof surfaces. The roof surface can be used for resting and viewing. Finally, we calculate the standard deviation of the roof surface area as a measure of the difference between all roof surfaces. To prevent the design from creating oversized roofs, we award the design with a smaller standard deviation value.

As a result, by integrating the factors included in the first two stages, the fitness function of the third stage is shown in Fig. 5, where *p_entrance* is the penalty function for the entrance space, *num_roof* is the number of roof surfaces, and the *view* is the average value of the unobstructed view on the surface of the building massing, *p_roof_area_diff* indicates the penalty function that punishes the design with a large difference of roof surface areas.

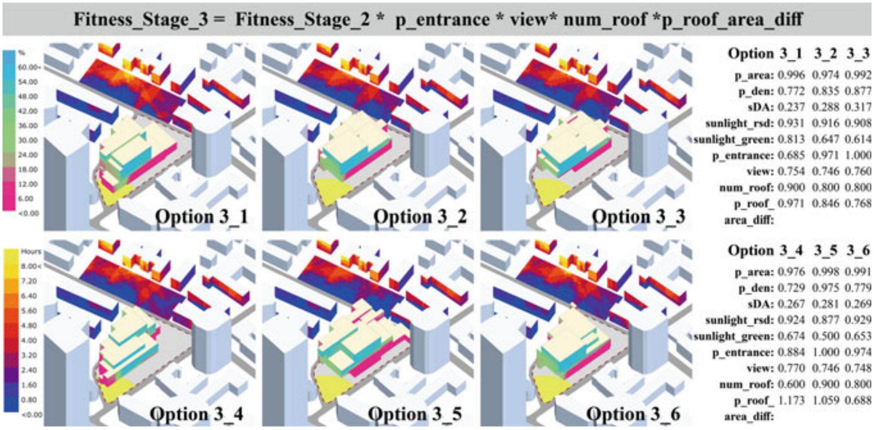

Fig. 5. The fitness evaluation function (top) and the optimization result in the third stage (bottom)

As shown in Fig. 5 (bottom), the optimal design options tend to place the major building massing volume in the middle of the site, which is the result of the compromise under multiple constraints. First, the building massing volume needs to make room for the entrance space in the south, and some design options feature an overhanging block above the entrance. Second, the building massing volume still maintains a distance from the residential building and green space in the north to mitigate the sunlight obstruction. Third, the building massing volume also stays away from the high-rise buildings in the south to enhance the unobstructed view. Fourth, the building massing volume tends to have more roof surfaces at different heights to increase the rooftop area. Finally, the area distribution of the roof surface becomes more evenly distributed, making the building more balanced visually. However, with the inclusion of the new soft constraint, the optimization has to make more compromises on other design aspects, which is evident by the drop in other optimization metrics.

4 Discussion and Conclusion

The presented case study shows that the use of the computational design help designers identify implicit design constraints and problems, and thereafter, further stimulates an iterative design exploration process. Regarding the proposed workflow, the use of soft constraints facilitates the designer to embed their design intents into the optimization, while the weight-product approach allows the new design intents can be integrated with the existing ones. Hence, the pre-existing design implication can be preserved but also dialed down with new constraints included.

Figure 6 summarizes the numerical change of the optimization result regarding different optimization metrics during the three stages. We select the best 15 designs from each design stage and calculate the average value for each metric. Left of the dotted line is the optimization objective included in the corresponding stage. At the same time, we also evaluate the optimal design options found in the earlier stage against the optimization metric included in the later stage to provide a holistic view of the change in optimization metrics. To compare the change in the value, we use the metric of the last stage as a benchmark (0.50) and recalculate the average value of the metric at earlier stages.

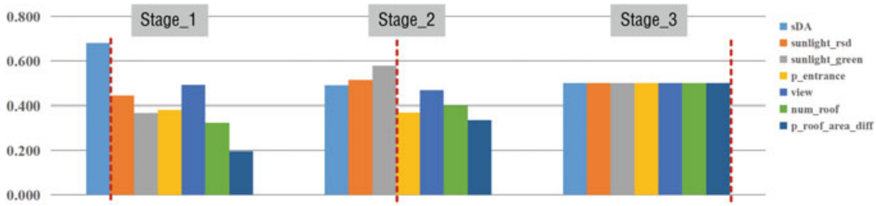

Fig. 6. The change in the value of each metric in the three stages

It is noticeable that the inclusion of each new metric in the optimization typically lowers the metric already in the optimization. This highlights that the design options produced in the final stage are the trade-off that achieves an acceptable compromise among various design aspects. Nevertheless, it should also be stressed that the design options produced in the final stage are more architecturally appealing and rational compared with those produced in earlier stages.

To conclude, this study is aimed to further extend the application of performance-based design optimization in architectural design and demonstrates how subjective design intents can be incorporated into the optimization process by using soft constraints. As demonstrated in the presented case study, the proposed design optimization workflow incorporating the application of soft constraints enables a more integrated human–computer design process. The optimization can effectively stimulate design exploration and assist designers in defining the design problem. Thus, by strengthening the feedback loop between designers and computers, a co-evolutionary design process emerges, where the application of performance-based design optimization provides designers with a "medium of reflection" in the early-stage design ideation and conceptual development.

Acknowledgements. This study is funded by China Postdoctoral Science Foundation (2021M701664) and National Natural Science Foundation of China (52178017).

References

1. Eiben AE, Smith JE (2003) Introduction to evolutionary computing, evolutionary computation.Springer Berlin Heidelberg (Natural Computing Series), Berlin, Heidelberg
2. Huang W, Xu W, Wang T (2011) Structural form generation using interactive genetic algorithm. In: Circuit bending, breaking and mending-proceedings of the 16th international conference on computer-aided architectural design research in Asia, CAADRIA 2011, pp 751–760
3. Marsault X (2013) A multiobjective and interactive genetic algorithm to optimize the building form in early design stages. In: Proceedings of building simulation, pp 809–816

4. Schön DA (1992) Designing as reflective conversation with the materials of a deisgn situation. Knowl-Based Syst 5(1):3–14
5. Shi X et al (2016) A review on building energy efficient design optimization from the perspective of architects. Renew Sustain Energy Rev 65:872–884. https://doi.org/10.1016/j.rser.2016.07.050
6. Wang L et al (2020) Algorithmic generation of architectural massing models for building design optimisation: parametric modelling using subtractive and additive form generation principles. In RE: anthropocene, design in the age of humans-proceedings of the 25th international conference on computer-aided architectural design research in Asia, CAADRIA 2020, pp 385–394
7. Wang L, Janssen P, Ji G (2018) Utility of evolutionary design in architectural form finding: an investigation into constraint handling strategies. In Gero JS (ed) Design computing and cognition '18. Springer, Cham, pp 177–194. https://doi.org/10.1007/978-3-030-05363-5_10
8. Wang L, Janssen P, Ji G (2020) SSIEA: a hybrid evolutionary algorithm for supporting conceptual architectural design. Artif Intell Eng Des Anal Manuf 34(4):458–476. https://doi.org/10.1017/S0890060420000281
9. Yousif S, Clayton M, Yan W (2018) Towards integrating aesthetic variables in architectural design optimization. In 106th ACSA annual meeting proceedings, the ethical imperative. ACSA Press, pp 430–436. https://doi.org/10.35483/ACSA.AM.106.68