Advantages of surrogate models for architectural design optimization

Thomas Wortmann, Alberto Costa, Giacomo Nannicini and Thomas Schroepfer

Artificial Intelligence for Engineering Design, Analysis and Manufacturing / Volume 29 / Special Issue 04 / November 2015, pp 471 - 481
DOI: 10.1017/S0890060415000451, Published online: 07 October 2015

Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2015), *29*, 471–481.

doi:10.1017/S0890060415000451

Advantages of surrogate models for architectural design optimization

THOMAS WORTMANN,[1] ALBERTO COSTA,[2] GIACOMO NANNICINI,[2] AND THOMAS SCHROEPFER[1]

[1]Architecture and Sustainable Design, Singapore University of Technology and Design, Singapore
[2]Engineering Systems and Design, Singapore University of Technology and Design, Singapore

(Received September 15, 2014; Accepted April 9, 2015)

Abstract

Climate change, resource depletion, and worldwide urbanization feed the demand for more energy and resource-efficient buildings. Increasingly, architectural designers and consultants analyze building designs with easy-to-use simulation tools. To identify design alternatives with good performance, designers often turn to optimization methods. Randomized, metaheuristic methods such as genetic algorithms are popular in the architectural design field. However, are metaheuristics the best approach for architectural design problems that often are complex and ill defined? Metaheuristics may find solutions for well-defined problems, but they do not contribute to a better understanding of a complex design problem. This paper proposes surrogate-based optimization as a method that promotes understanding of the design problem. The surrogate method interpolates a mathematical model from data that relate design parameters to performance criteria. Designers can interact with this model to explore the approximate impact of changing design variables. We apply the radial basis function method, a specific type of surrogate model, to two architectural daylight optimization problems. These case studies, along with results from computational experiments, serve to discuss several advantages of surrogate models. First, surrogate models not only propose good solutions but also allow designers to address issues outside of the formulation of the optimization problem. Instead of accepting a solution presented by the optimization process, designers can improve their understanding of the design problem by interacting with the model. Second, a related advantage is that designers can quickly construct surrogate models from existing simulation results and other knowledge they might possess about the design problem. Designers can thus explore the impact of different evaluation criteria by constructing several models from the same set of data. They also can create models from approximate data and later refine them with more precise simulations. Third, surrogate-based methods typically find global optima orders of magnitude faster than genetic algorithms, especially when the evaluation of design variants requires time-intensive simulations.

Keywords: Computer-Aided Design Tool; Design Optimization; Design Space Exploration; Radial Basis Function Method; Surrogate-Based Optimization

1. INTRODUCTION

Climate change, resource depletion, and worldwide urbanization feed the demand for more efficient buildings, that is, better-designed buildings. However, architectural design is notoriously complex, mainly because evaluation criteria can be contradictory and often change during the design process (e.g., Rittel & Webber, 1973). Woodbury and Burrow (2006) characterize design as an exploratory activity that uncovers design variants in a design space. Because design is time consuming, designers typically explore only small portions of a design space. Architects have responded to the demand for more resource- and energy-effective buildings by calling for a more "performative" architecture (e.g., Kolarevic & Malkawi, 2005; Hensel, 2013), and to the complexity of architectural design problems by proposing more performative design tools (e.g., Kolarevic, 2005; Shea et al., 2005; Oxman, 2008; Shi & Yang, 2013). These design tools often involve parametric modeling (Woodbury, 2010), simulations (Malkawi, 2005), and sometimes optimization (Shi & Yang, 2013; Lin & Gerber, 2014). Designers can generate many design variants with parametric modeling, and evaluate them with structural and environmental simulation packages. However, sophisticated simulations often take a long time, even on

Reprint requests to: Thomas Wortmann, Architecture and Sustainable Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. E-mail: thomas_wortmann@mymail.sutd.edu.sg

today's computers. Optimization methods search strategically for design variants that minimize or maximize a predefined performance measure. In other words, optimization methods predict which design variants are the most promising to evaluate, and thus allow faster and more extensive design exploration. Ultimately, optimization can thus contribute to a better ecological performance of the built environment.

Beyond advocating genetic algorithms, a randomized optimization method inspired by evolution (see Holland, 1992), architects have reflected little on what optimization methods are most appropriate for architectural design. However, applying optimization methods to architectural design is not at all straightforward: many architectural design theorists stress the complex, nonquantifiable, or even "wicked" nature of design problems, and the nonlinear and subjective character of design processes (e.g., Rittel & Webber, 1973; Lawson, 2006; Gänshirt, 2007). Genetic algorithms provide no insights into a design problem beyond proposing specific alternatives with above average, and sometimes close to optimal, performance values. In addition and perhaps consequently, architectural designers rarely use optimization, despite extensive discussions in the literature. This paper presents surrogate-based optimization as an approach that is faster and better suited to the flexible character of architectural design problems. After discussing current methods in architectural design optimization (Section 1.1) and the challenges of integrating optimization methods into the architectural design process (Section 1.2), the paper introduces surrogate-based optimization (Section 2), presents two applications from daylight optimization (Section 2.1), and discusses the advantages of surrogate-based methods over other kinds of optimization (Section 2.2).

1.1. Optimization problems and methods in architectural design

Optimization problems in architectural design are complex and have multiple objectives. In structural optimization, for example, an optimal configuration minimizes structural displacement as well as the structure's dead load. In this paper, we optimize a building façade according to the competing criteria of maximizing daylight and minimizing glare.

This complexity of evaluation criteria often results in nonconvex fitness spaces with several local optima. Because of this nonconvexity, conventional, gradient-based optimization is widely regarded as unsuitable for architectural design. Instead, architectural designers approach optimization problems with methods that assume no prior knowledge of the problems' mathematical formulation. In the optimization community, such methods are known as "blackbox" or "derivative-free" (Conn et al., 2009). These methods roughly fall into two categories: surrogate-based methods and direct-search methods (see Kolda et al., 2003).

Metaheuristics such as genetic algorithms are a third type of blackbox methods. Metaheuristics are randomized algorithms and often biologically inspired (Yang, 2010). Genetic algorithms have been advocated as especially appropriate for the architectural field (De Landa, 2002; Oxman, 2008; Miles, 2010). They dominate the literature on optimization in architectural design and have proven suitable for a wide range of architectural design problems (e.g., Kicinger et al., 2005; Lin & Gerber, 2014). Grasshopper®, a free parametric design plugin for the popular three-dimensional modeling application Rhinoceros®, includes an easy-to-use evolutionary solver named Galapagos (Rutten, 2013). Simulated annealing (also implemented in Galapagos) and particle swarm optimization are other metaheuristics that are sometimes used in the architectural field (e.g., Luebkeman & Shea, 2005; Felkner et al., 2013). According to Yang (2010), metaheuristics are also widespread in engineering design.

Two key advantages probably account for this popularity: first, metaheuristics usually are much easier to implement than direct-search and surrogate-based methods; and second, metaheuristics are applicable to almost any kind of optimization problem, regardless of the type (e.g., continuous/discrete, linear/nonlinear, convex/nonconvex) and number of variables. Nevertheless, the optimization community has remained skeptical. The authors of a text on derivative-free optimization regard heuristics as "methods of last resort . . . and would use them only if nothing better is available" (Conn et al., 2009, p. 6).

A major reason for this skepticism is that metaheuristics often have poor convergence properties, relying on randomization in the hope of finding a global optimum. In contrast, Gutmann (2001) has shown that the surrogate-based optimization method presented in this paper deterministically converges to the global optimum under mild assumptions, if given sufficient time (cf. Costa & Nannicini, 2014). Even though many metaheuristics eventually find an optimum, on practical problems other blackbox methods typically show much faster convergence toward the global optimum (cf. Regis & Shoemaker, 2007). Such faster convergence is evident in our second case study, which compares the performance of a genetic algorithm with our surrogate-based optimization method (see Section 2.1.2). While fast convergence may not be of paramount importance when evaluating the fitness of the design candidate is computationally inexpensive, as is typically the case for the problems that metaheuristics are applied to, many practically relevant situations (such as the one discussed in this paper) greatly benefit from faster convergence.

1.2. Challenges for optimization in the architectural design process

Despite large amounts of academic literature devoted to the subject, optimization methods have had a limited impact on the architectural profession. This lack of popularity is especially apparent in comparison with engineering design, where optimization methods have become relatively commonplace (Flager & Haymaker, 2009). This paper identifies two potential reasons for this reluctance among architectural designers: time intensity and, perhaps of the most importance, the complexity of architectural design.

Optimization is time intensive, especially when the problem involves structural or environmental simulations. Genetic algorithms often require many iterations of such simulations. This time intensity motivates the search for alternative optimization techniques, such as particle swarm optimization (e.g., Hassan et al., 2005; Felkner et al., 2013).

Beyond the issue of speed, design theorists have questioned the applicability of optimization techniques for architectural design in general. According to Rittel and Webber (1973), optimization has to rely on predefined parametric models and performance criteria and is thus difficult to integrate into nonlinear design processes with changing performance criteria:

> The methods of Operations Research [. . .] become operational, however, only after the most important decisions have already been made, i.e., after the problem has already been tamed. Take an optimization model. Here the inputs needed include the definition of the solution space, the system of constraints, and the performance measure as a function of the planning and contextual variables. But setting up and constraining the solution space and constructing the measure of performance is the wicked part of the problem. Very likely it is more essential than the remaining steps of searching for a solution which is optimal relative to the measure of performance and the constraint system.

In other words, to make optimization possible, the designer has to turn a complex design problem into a well-defined one by formulating a parametric model and quantitative criteria for evaluating design variants. Similarly, in Lawson's characterization (Lawson, 2006, pp. 121–122), design problems are inherently complex, and therefore necessarily require the designer's subjective judgment:

> Rarely can the designer simply optimize one requirement without suffering some losses elsewhere. Just how the trade-offs and compromises are made remains a matter of skilled judgement.

As is shown below in Section 2.2.4, surrogate-based optimization addresses the inherent complexity of design problems by significantly reducing the effort for evaluating the same data according to different evaluation criteria. In this way, a designer can illuminate the trade-offs and compromises that characterize a design problem.

2. SURROGATE-BASED OPTIMIZATION AND THE RADIAL BASIS FUNCTION (RBF) METHOD

This paper proposes that, compared to metaheuristics such as genetic algorithms, surrogate-based methods are better suited for design space exploration in the architectural field. Such methods have already been successfully applied in engineering design (e.g., Björkman & Hölmstrom, 2000, Johan & Wojciechowski, 2007, Hemker, 2008).

Metaheuristics approach the design space as a blackbox and evaluate isolated variants in this space according to various search strategies. Although evaluating isolated design variants can lead to good and in some cases optimal results, such a search does not contribute to an increased understanding of the design space as a whole. In other words, metaheuristics explore the design space only insofar as they evaluate specific design variants. In practice, such methods usually do not even present all the variants that they have evaluated. Rather, they present a selection of the best variants, because, for a human designer, the large amount of "randomly" chosen variants does not provide an additional understanding of the design space.

In contrast, surrogate models construct a simplified mathematical model from the evaluated variants and use this model to guide further exploration (e.g., Koziel et al., 2011). To construct this model, a so-called response surface is interpolated between a set of strategically chosen design variants. In other words, the response surface is a mathematical approximation of the design space, as defined by the parameters of the optimization problem.

Wang and Shan (2006) identify three approaches to design optimization with surrogate models: construct a surrogate model, and use an optimization algorithm to look for an optimum in the model; construct a surrogate model with a sampling method embedded in an (iterative) optimization algorithm; and construct a surrogate model and look for an optimum (iteratively) without employing an explicit optimization algorithm.

The method presented here falls into the second category: the selection of the optimal design variant is an iterative process of evaluating a new variant, constructing an improved model based on this and the previous evaluations, and then choosing the next point to evaluate based on this improved model. Every evaluation of new variants thus contributes to an improved model of the design problem. This type of algorithm tries to balance the need to optimize (i.e., to guide the search toward a design variant with a maximum, or at least very good, performance value) and the need to improve the model of the performance measure (i.e., to identify and explore the areas of the design space associated with a high model uncertainty). The process stops after a fixed number of evaluations, or when it reaches the desired accuracy of the surrogate model.

Another important difference between metaheuristics and surrogate-based methods is that the latter are typically applied on problems with a limited number of variables. However, many problems from architectural design can be expressed using relatively small-sized mathematical formulations. Whereas surrogate-based methods initially dealt with continuous variables only (cf. Gutmann, 2001), later advances allow the treatment of integer variables as well. In Section 2.1.2, we present a successful application of our implementation of a surrogate model method to a daylight optimization problem with 15 integer variables.

Other methods, such as direct search, also can solve blackbox problems. The key difference between surrogate-based and direct-search methods is that the former construct global models of the unknown performance measure, while the latter build only partial, local models to guide local improvement steps.

Koziel et al. (2011) list several methods for creating surrogate models. Simpson et al. (2001) and Wang and Shan (2006) provide comprehensive surveys of methods from the perspective of engineering design. Among the most well known are Kriging methods such as EGO (Jones et al., 1998), the original RBF method of Gutmann (2001), and the stochastic RBF method (Regis & Shoemaker, 2007). Mullur and Messac (2005) have developed an extension to RBF models. Other methods include support vector regression (Smola & Schölkopf, 2004) and NURBS (Turner et al., 2007).

The application and computational tests presented below are based on the RBF method as implemented in the RBFOpt library (Costa & Nannicini, 2014). Compared to other surrogate-based approaches, RBF methods are especially effective for engineering problems that involve time-intensive simulations (Holmström et al., 2008). The next section introduces two applications of surrogate models, and of the RBF method in particular, to architectural daylight optimization.

2.1. Two daylight optimization case studies with the RBF method

As first case studies for the integration of the RBF method into architectural design processes, we apply the method to two different optimization problems related to the design of a mixed-use high-density church building in Singapore, New Jurong Church. The church has a performative façade that modulates daylight conditions on the interior due to the differentiated inclinations of its small, disk-shaped louvers (see Fig. 1). Formulated as an optimization problem, the values of the angles of the louvers are the design parameters or variables, and daylight quality (defined below) is the evaluation criterion. We associate one design variable with the opening angle of a group of louvers, instead of individual louvers. This simplification serves three purposes: it reduces the dimension of the optimization problem, guarantees a more unified visual effect of the façade, and standardizes the façade elements for easier construction. For an individual room inside the building, lighting conditions depend only on the part of the façade adjacent to that room. Accordingly, in the following, we use RBFOpt to find the optimal configuration of the disk-shaped louvers adjacent to two individual rooms.

Every design variable thus represents the opening angle of a group of louvers, and takes values between 0° (louvers closed) and 180° (louvers fully open). Daylight is simulated with DIVA, a plugin for the three-dimensional modeling software Rhinoceros (Jakubiec & Reinhart, 2011) that yields illuminance values inside the building. In our case studies, which employ reduced mesh geometries with around 8000 mesh faces, the annual daylight and glare simulations take about 5 min each, using an Intel Dual Core i5-4200M 2.5-GHz CPU.

2.1.1. *Optimizing useful daylight illuminance and glare with one variable*

For the first optimization problem, we assume that all the louvers along a room's façade have the same angle, expressed as one continuous variable from 0° to 180°. Our goal is to find the range of angles that result in good lighting conditions on the interior of a single room. We use this simple single-variable optimization problem as an illustrative example to provide an intuition of the idea behind surrogate-based methods. In the next section, we apply the method to a more realistic case study with more design variables.

Fig. 1. Impression of the performative façade.

For optimization problems, we need to define rigorous evaluation criteria in the form of a so-called objective function. In other words, given an output of the lighting simulation, how can we assess its quality? We assess the quality of lighting conditions with two metrics calculated in DIVA. The first one is useful daylight illuminance (UDI). UDI is defined as the annual occurrence of illuminances across the work plane that are within a range considered "useful" by occupants (Mardaljevic et al., 2012). In a survey of occupant preferences and behavior in daylit offices, occupants considered daylight illuminances in the range of 100 to 300 lux effective, either as the sole source of illumination or in conjunction with artificial lighting. They perceived daylight illuminances in the range 300 to around 3000 lux as desirable or at least tolerable. Values above 3000 lux are associated with both glare and excessive warmth due to solar gains.

In this first optimization case study, we combine UDI with an annual glare metric. This glare metric is an average of values in the range [0, 1], with each value representing the daylight glare probability (DGP) for a given hour and day of the year (from a specific viewpoint). We map these values to a trapezoidal function such that a value of 1 indicates that there is no or only imperceptible glare (DGP < 0.35), a value of 0 indicates intolerable glare (DGP ≥ 0.45), and in-between values indicate the amount of noticeable glare. (See Fig. 2 for a graphic representation of glare values over 1 year.)

We define the objective function in terms of UDI and glare. Let s be the output of a lighting simulation. Let $U(s)$ be the value of UDI normalized to be in the range [0, 1], $G(s)$ be a value in the range [0, 1] expressing the average quality of the glare values over 1 year, and $T(s)$ equal to 1 if there are any intolerable glare values during the year, and 0 otherwise. The objective function (from Costa et al., 2015) we want to maximize is

$$f(s) = \alpha U(s) + (1-\alpha)G(s) - \alpha(1-\alpha)|U(s) - G(s)| \\ - T(s), \quad \alpha \in (0, 1).$$

The parameter α decides the relative weight of glare and UDI for the considered room, which is a decision taken by the designer. The term $T(s)$ insures that solutions that allow intolerable values of glare result in a negative (i.e., infeasible) value of the objective function. The term $\alpha(1-\alpha)|U(s) - G(s)|$ penalizes solutions that present good values of UDI but bad values of glare, or vice versa. The values of $f(s)$ are in the range $[-1, 1]$, and negative solutions are assumed infeasible, that is, unacceptable in practice. Notice that $f(s)$ is not defined in terms of the design parameters (i.e., the angles of the disk-shaped louvers), but is computed from the output of the daylight simulations. The objective function $f(s)$ is the performance measure that we are trying to maximize.

To construct a surrogate model, the algorithm requires the values of the performance measure for several parameter values. In this example, for a room with a single, south-facing facade, we simulated UDI and glare for angles of 0°, 30°, 60°, 75°, 90°, 105°, and 180°. The model calculated by RBFOpt predicts an optimum value of around 130° (see Fig. 3). However, as discussed in Section 2.2.2, this model not only predicts the optimum value but also serves as a tool for design space exploration by approximating the relationship between design parameters and expected daylighting quality for the global design space.

2.1.2. *Optimizing UDI with 15 variables*

For the second optimization problem, we consider a south-facing chapel of the church. The chapel has a 9-m long façade toward the southeast and a 6-m long façade to the southwest. In total, the 2 façades consist of 15 façade panels, each with a group of 462 openings with disk-shaped louvers. The problem assumes that every panel has an individual angle for its group of louvers, with angles from 0° to 180° available in increments of 5° (to allow standardization).

In other words, the problem has 15 design variables (1 for the louver angle of each panel), and every variable can take one of 37 integer values (0°, 5°, 10°, . . . , 170°, 175°, 180°). The problem's objective is to maximize UDI, while maintaining a visually comfortable and aesthetically pleasing configuration of the louvers. For this problem, we do not explicitly simulate glare. Instead, by counting only lux values

Fig. 2. Glare values over 1 year for a room of the church in Singapore, according to the date and time of day.

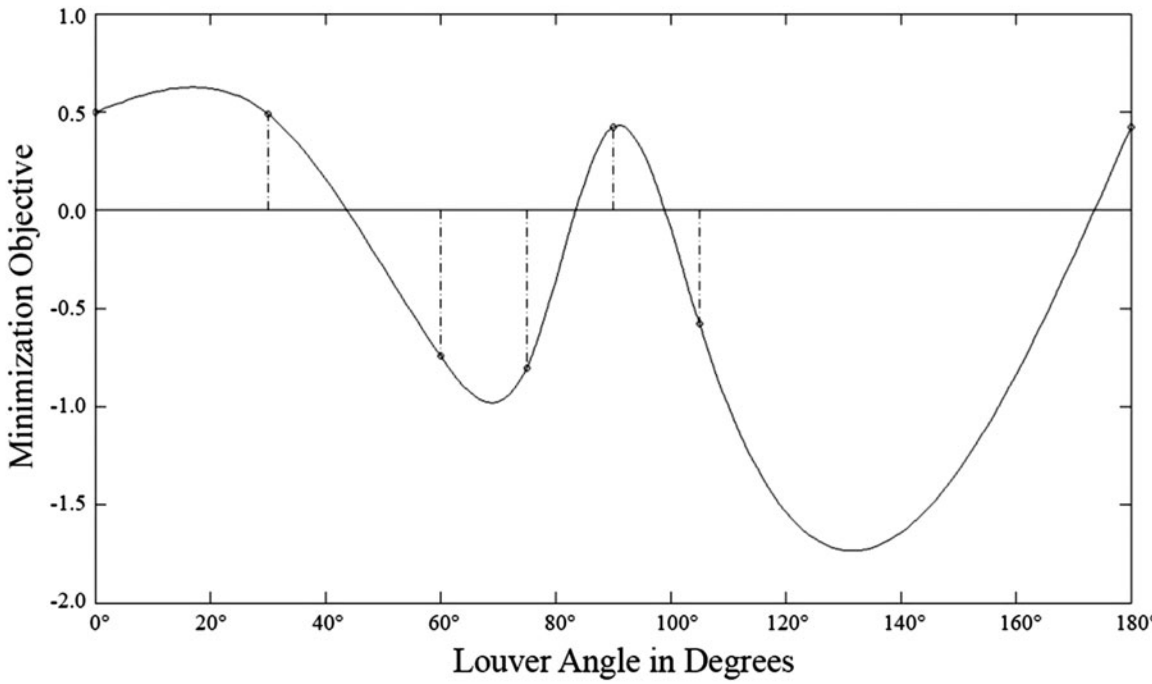

Fig. 3. Surrogate model based on the simulation output for seven values of angles of the disk-shaped louvers. Note that the problem is formulated as a minimization problem (i.e., good solutions are those for which the model assumes small values).

below 3000 toward the objective, we implicitly avoid solutions with intolerable glare.

Because, for reasons of visual comfort and aesthetics, the outside appearance of the façade and the interior brightness of the room should change only gradually, we penalize differences between neighboring façade panels that are larger than 10°. For every panel, we define a squared error e_i, which we compute from the angles d_i and d_{i-1} of each panel and its left neighbor:

$$e_i(s) = ((|d_i - d_{i-1}| - 10^\circ)/170^\circ)^2.$$

The average sum of the 15 error terms yields an overall error term $E(s)$, multiplied by a weight α that modifies the importance of gradual angle changes relative to UDI. (We chose α to be 100.) The objective thus consists of maximizing UDI, which we simulate, while avoiding large angle differences between neighboring panels:

$$f(s) = U(s) - \alpha E(s).$$

We ran this problem with both RBFOpt and Galapagos, the genetic algorithm included in Grasshopper. The experiment was repeated five times, with each algorithm running for 100 iterations. In Galapagos, we ran four generations with a population size of 25. Besides the iteration limit and the population size in Galapagos, we used default settings for both algorithms.

On average, after 100 evaluations, RBFOpt finds an objective value of 0.78 for the maximization problem, while Galapagos finds a value of 0.05. In three of the five runs, Galapagos did not achieve an improvement over its initial population. Out of the five runs, the best configuration found by RBFOpt has an objective value of 0.83 with a UDI of 86% and a (weighted) penalty for error differences of 0.03, and the best configuration Galapagos found has an objective value of 0.16 with a UDI of 24% and a (weighted) penalty for error differences of 0.08 (also see Figs. 4 and 5).

The 100 evaluations of each run require 100 simulations and therefore approximately 8 h of computing time. One generation in Galapagos (with 25 evaluations) takes about 2 h to compute. Given that genetic algorithms typically about take many generations, and hundreds and sometimes thousands of evaluations, to converge, it seems that Galapagos would require a prohibitive amount of computing time to find a good design candidate for problems that require detailed simulations, such as the one studied in this section. In contrast, in our tests, the surrogate-based method finds good design candidates after relatively few evaluations.

2.2. Advantages of surrogate-based optimization in architectural design

Compared with metaheuristics, surrogate-based optimization methods have important advantages for architectural design. Surrogate-based methods find solutions faster, allow the designer to explore quickly an approximation of the solution space, and can be refined progressively. In addition, a designer can recalculate the surrogate model essentially in real time to explore the impact of different evaluation criteria (i.e., objective functions) on the solution space.

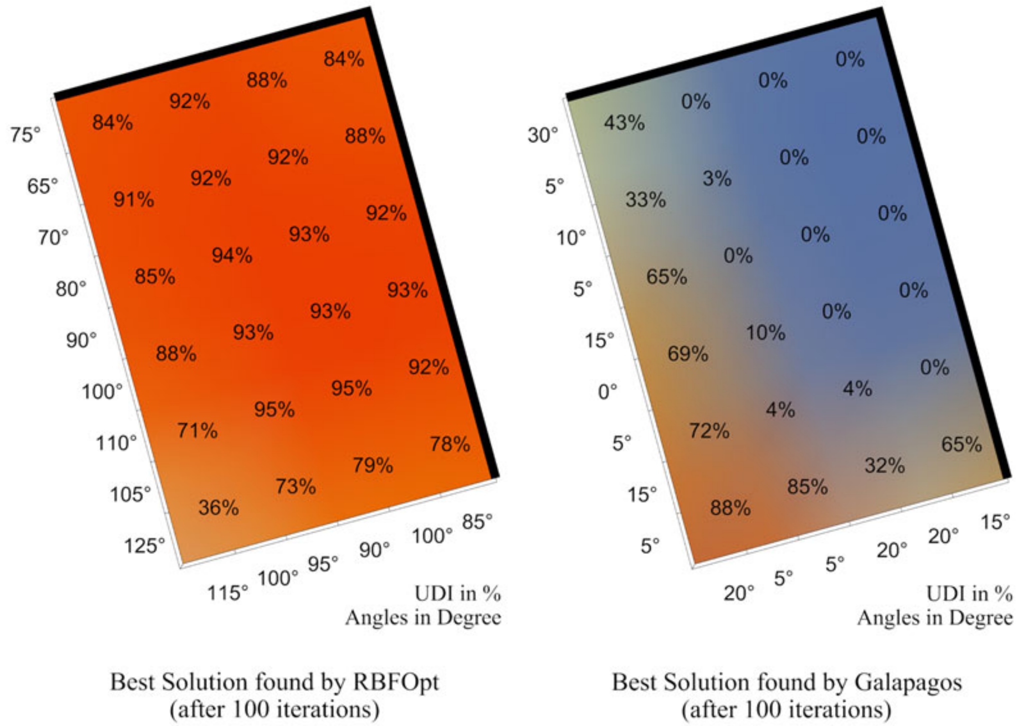

Fig. 4. Comparison between the best solutions found by RBFOpt and Galpapagos. The plan diagrams indicate the percentage of useful daylight illuminance predicated at the simulation nodes on the interiors, and the louver angles of the panels around the facades.

2.2.1. Speed

On the daylight optimization problem with 15 integer variables discussed in Section 2.1.2, RBFOpt significantly outperforms Galapagos, an evolutionary solver. This result confirms a trend of benchmark results from the optimization community.

In a comparison between different implementations of optimization algorithms, DAKTOA/EA, an implementation of various genetic algorithms did relatively poorly (Rios & Sahinidis, 2013). Similarly, Holmström et al. (2008) conclude that for blackbox, nonconvex optimization problems with constraints "deterministic derivative-free methods [such as the RBF method] compare well with the derivative-based ones, but the stochastic genetic algorithm solver is several orders of magnitude too slow for practical use."

The impression that, compared to other derivative-free methods, genetic algorithms are significantly slower is confirmed by benchmark results reported by Costa and Nannicini (2014). As shown in Table 1, RBFOpt outperforms other black-box algorithms on several test functions from the literature.

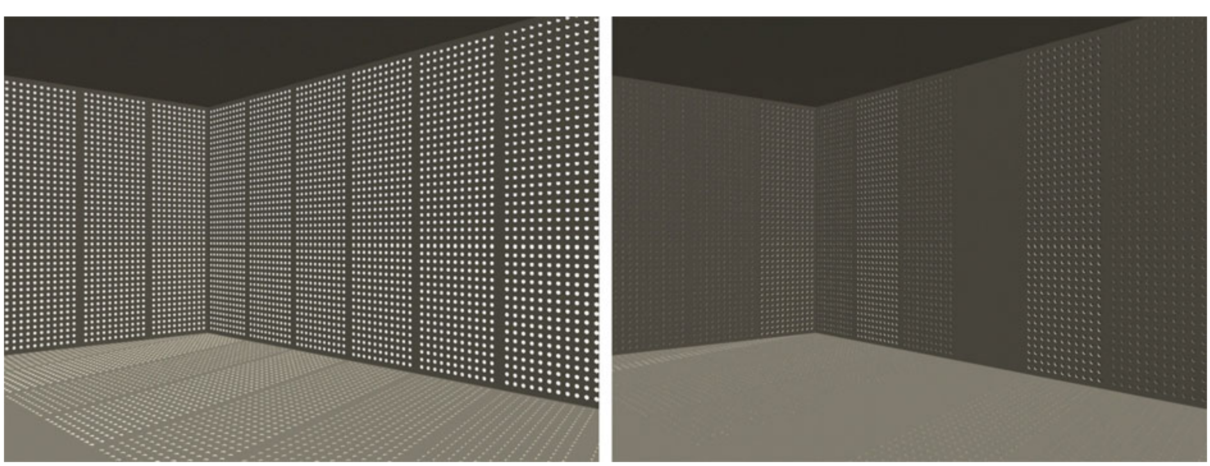

Fig. 5. Comparison between visual impressions of the best solutions found by (left) RBFOpt and (right) Galpapagos. Note that the appearance of the façade found by RBFOpt is more uniform.

Table 1. *Number of evaluations to find a value within 1% from the global optimum for different blackbox methods on test functions from the literature*

Test Function	Dimension	RBFOpt	DIRECT	GLOBAL –QN	DE
Branin	2	30.5	63	77	1190
Goldstein–Price	2	52.5	101	277	1018
Hartman 3	3	44.5	83	196	476
Shekel 10	4	83.46	97	2378	6194
Hartman 6	6	92.82	213	703	7220

Note: A lower number indicates faster convergence. Dimension, the number of continuous variables; RBFOpt, surrogate model method; DIRECT, direct search method; GLOBAL–QN, stochastic surrogate model method; DE, genetic algorithm. For RBFOpt, we report the averages over 20 iterations. For GLOBAL, we report the results obtained with the best configuration, that is, a quasi Newton (QN) method for local search.

Specifically, RBFOpt outperforms the popular genetic algorithm Differential Evolution (see Storn & Price, 1997) by almost two orders of magnitude. RBFOpt is also faster than DIRECT, a direct-search method (see Jones et al., 1998), and GLOBAL, a stochastic surrogate-based method (see Csendes et al., 2008).

Note that the benchmarks above compare the number of evaluations of the blackbox function, in our case, the number of actual simulations. From the perspective of these benchmarks, the surrogate model is a by-product of the optimization process.

2.2.2. *Exploring the (approximate) solution space*

According to their characterization of architectural design as design space exploration, Woodbury and Burrow (2006) highlight the need for computational design tools that support this exploration. In contrast to metaheuristics, which examine only a randomized set of design variants, surrogate models approximate the mathematical model implied by the parametric model and the evaluation criteria. With a surrogate model, designers can approximate how changed design parameters affect a design's performance without the need for additional, time-intensive simulations. A surrogate model can be understood as a rough mapping of the "solution space" of a design problem and, in that sense, as a manifestation of design knowledge (see Lawson, 2004). Extending Woodbury and Burrow's (2006) definition of the design space as the space containing all design variants, this solution space not only contains all design variants but also has an additional dimension that indexes a design variant's performance. In this way, surrogate models serve as design tools that approximately but quickly evaluate the expected performance of different design variants.

For example, the result of the optimization process with one variable described in Section 2.1.1 is a mathematical model that, given a room's orientation, relates the opening angle of the louvers to the expected daylight performance. This model indicates that, for a south-facing room, the performance is best for angles around 130° (see Fig. 3). However, from a design standpoint, the observation that good performance is also possible for angles ranging from around 40° to 80° is more important. This insight opens up considerable freedom for a designer to choose angles according to other, perhaps nonperformance-related, criteria. (For example, a designer might desire a variation of angles for aesthetic reasons. Alternatively, certain angles might be easier to manufacture.)

In other words, surrogate models not only propose good solutions but also contribute to an improved understanding of the design problem by facilitating playful interactions between the designer and the mathematical formulation of the design problem. Lawson (2006, p. 282) characterizes the integration of computers into the design process as a "conversation," where "the greatest responsibility and certainly the final say will rest with the human designer." Surrogate models aid in engaging designers in such conversations by relating design parameters to evaluation criteria without prescribing an "optimal" solution. In a concrete example, Kolarevic (2005) proposes to animate parametric models in relation to performance criteria, "from the given condition to the optimal condition, with the assumption that the designer could find one of the in-between conditions interesting and worth pursuing, even though it may not be the most optimal solution." Surrogate models are a way to predict performance in such a scenario. (In addition to referencing a surrogate model, a designer should explicitly evaluate promising design candidates, because a surrogate model does not offer any guarantee of accurately predicting the performance of design candidates.)

Of course, one cannot expect that the approximation given by the surrogate model is always accurate. However, Costa and Nannicini (2014) show that it is possible to compute a "confidence index" to obtain an indication of the quality of the surrogate model. When this index satisfies a certain criterion, they empirically show that on a large set of mathematical models, the surrogate model is effective at predicting the change in the performance measure when varying one of the design parameters. In particular, in their tests, variations in the value of design parameters by no more than 5% of the parameter range yield predictions with a relative error smaller than 50% roughly 80% of the time, and 60% of the time the error is no more than 5%. For smaller variations, the accuracy improves.

This area is currently under research. From the perspective of design space exploration, surrogate models provide only an imperfect map of the solution space. However, even an imperfect map is better than none, especially in early phases of the design process. As is discussed in the next section, a designer can iteratively improve the accuracy of the surrogate model while developing the design based on the model's imperfect, yet informative, estimates.

2.2.3. *Refining the (approximate) solution space*

Another appealing property of the RBF method from the standpoint of design exploration is that it allows progressive refinement of the surrogate model. A relatively small number of data points (i.e., simulations) suffices to build an approximate model of the solution space (more precisely, at least as many evaluations as design variables are needed). Increasingly precise data points can iteratively improve this model. In this way, the designer can quickly build "rough" models that can be refined as more information becomes available during the design process.

When evaluating new design variants, the algorithm balances the "exploration" of unknown parts of the solution space and the "exploitation" of the surrogate model, which is used to guess the location of the optimal solution. The more design candidates are evaluated, the better the quality of the surrogate model. It can be mathematically shown that the surrogate model of the performance measure and the actual performance measure eventually coincide (Gutmann, 2001). In practice, the quality of the surrogate model at a given iteration of the algorithm can be estimated using cross-validation techniques (Costa & Nannicini, 2014).

Recent developments of the RBF method offer the appealing possibility to exploit simulations with different levels of precision. For lighting simulation, changing the accuracy of the simulation is typically possible. Costa and Nannicini (2014) propose to build an approximate model using the output from low-accuracy, hence faster, simulations, and to perform high-accuracy simulations only for those design variants that are potentially optimal or at least of very good value. However, this approach requires designers to judge the relative precision of the low- and high-accuracy simulations.

Cassioli and Schoen (2013) demonstrate another way in which design expertise can hasten the convergence of a surrogate-based optimization process. In particular, they report that *a priori* knowledge of some (lower or upper) bounds on the possible values of the performance measure can yield a reduction in the number of design candidates that need to be evaluated. For example, the optimum in the model presented in Figure 3 goes beyond what we define as possible for the objective function. (The optimum is around –1.75, with the range of the objective function defined from –1 to 1.) Including such information in the optimization process can lead to faster convergence. In this way, the knowledge a designer already possesses about a design problem (for example, through prior experience with similar problems) can contribute to a better optimization result.

2.2.4. *Changing the objective function*

Designers can construct surrogate models with different objective functions without repeating time-intensive simulations, which is another property that makes them good candidates for design space exploration. In many situations, such as daylight simulation, one can save the (detailed) output data of a simulation run to allow the evaluation of a different objective function with already available output data. In this way, one quickly obtains a model according to the modified performance criteria, and can run the optimization method with the already existing data. This reuse of existing results leads to much faster convergence than restarting from scratch, because the method is initialized with all the design candidates previously explored, now evaluated according to a different performance measure.

Based on this rapid convergence, the designer can gauge the impact of different evaluation criteria on the design by exploring a range of different solution spaces. In other words, designers can interact not only with one surrogate model, as described in Section 2.2.2, but with a range of models, or space of solution spaces, that reflect different potential evaluation criteria. This possibility directly addresses the challenge that "constructing the measure of performance is the wicked part of the problem" (Rittel & Webber, 1973). In other words, the opportunity to interact with different surrogate models of the same design problem facilitates the "skilled judgment" that, according to Lawson (2006, p. 122), is necessary for complex design problems such as architecture.

The objective function of the daylight optimization case study described in Section 2.1.1 combines two approaches for dealing with multiple objectives. First, one can combine different objectives with a weighted average, as is the case for UDI and glare. Second, one can ensure that design candidates meet certain constraints by penalizing their violations. We employ this approach to exclude design candidates with intolerable glare. In this manner, one objective can express different design considerations.

In such a case, much depends on the formulation and parameters of the objective function. With a surrogate-based optimization method, a designer can change the relative importance of brightness versus glare (for example, to accommodate a design change in terms of a room's function), or increase the penalty for angle changes between neighboring façade elements (to ensure efficient construction) and build surrogate models based on the new evaluation criteria in essentially real time. The designer can thus explore different design scenarios with a minimal need for time-intensive computations. This flexibility is key in the architectural design process, which typically involves not only changes to the solution but also changes to the design objectives.

3. CONCLUSION AND FUTURE RESEARCH

Surrogate-based optimization is a "designerly" optimization method that converges significantly faster than the metaheuristics currently employed by architectural designers and, more important, promotes design freedom. As shown in our case study on daylight optimization, surrogate models not only provide good solutions but also contribute to a better understanding of design problems. The possibility to examine the same data set with different evaluation schemes mitigates the inherent "wickedness" of design problems. In other words, surrogate models are a promising starting point for the development of exploratory design tools that enable conversations between designers and the computer, and help to generate and visualize design knowledge in the form of mathematical models.

The presented case studies have only partially explored the potential of surrogate models as design tools. To examine the advantages of surrogate models further, more case studies and benchmarks from different domains, such as structural design and multidisciplinary optimization, are needed. Another important question is how one can better integrate available knowledge about design problems into the optimization process. Further avenues of development are a user-friendly implementation and a visual interface that facilitates the exploration of the (space of) solution spaces.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their fruitful comments, which helped us to improve the paper. The research on New Jurong Church was started at Harvard University in collaboration with Christoph Reinhart, Alstan Jakubiec, and Azadeh Omidfar. New Jurong Church is a design project by Schroepfer + Hee. The daylight optimization study was supported by Singapore University of Technology and Design–MIT International Design Centre Grant IDG21300102.

REFERENCES

Björkman, M., & Hölmstrom, K. (2000). Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering 1(4)*, 373–397.

Cassioli, A., & Schoen, F. (2013). Global optimization of expensive black box problems with a known lower bound. *Journal of Global Optimization 57(1)*, 177–190.

Conn, A., Scheinberg, K., & Vicente, L. (2009). *Introduction to Derivative-Free Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Costa, A., & Nannicini, G. (2014). *RBFOpt: An Open-Source Library for Black-Box Optimization With Costly Function Evaluations*, Optimization Online No. 4538. Singapore University of Technology and Design.

Costa, A., Nannicini, G., Schroepfer, S., & Wortmann, T. (2015). Black-box optimization of lighting simulation in architectural design. In *Complex Systems Design & Management Asia* (Cardin, M.-A., Krob, D., Pao, C.L., Yang, H.T., & Wood, K., Eds.), pp. 27–39. New York: Springer.

Csendes, T., Pál, L., Sendín, J.O.H., & Banga, J.R. (2008). The GLOBAL optimization method revisited. *Optimization Letters 2(4)*, 445–454.

De Landa, M. (2002). Deleuze and the use of the genetic algorithm in architecture. In *Designing for a Digital World* (Leach, N., Ed.), pp. 117–118. London: Wiley.

Felkner, J., Chatzi, E., & Kotnik, T. (2013). Interactive particle swarm optimization for the architectural design of truss structures. *Proc. 2013 IEEE Symp. Computational Intelligence for Engineering Solutions (CIES)*. New York: IEEE.

Flager, F., & Haymaker, J. (2009). *A Comparison of Multidisciplinary Design, Analysis and Optimization Processes in the Building Construction and Aerospace*. Stanford, CA: Stanford University Press.

Gänshirt, C. (2007). *Tools for Ideas: Introduction to Architectural Design*. Basel: Birkhäuser.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization 19(3)*, 201–227.

Hassan, R., Cohanim, B., De Weck, O., & Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. *Proc. 1st AIAA Multidisciplinary Design Optimization Specialist Conf.* Reston, VA: AIAA.

Hemker, T. (2008). *Derivative free surrogate optimization for mixed-integer nonlinear black box problems in engineering*. PhD Thesis. TU Darmstadt.

Hensel, M. (2013). *Performance-Oriented Architecture: Rethinking Architectural Design and the Built Environment*, 1st ed. Chichester: Wiley.

Holland, J.H. (1992). Genetic algorithms. *Scientific American 267(1)*, 66–72.

Holmström, K., Quttineh, N.-H., & Edvall, M.M. (2008). An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. *Optimization and Engineering 9(4)*, 311–339.

Jakubiec, J.A., & Reinhart, C.F. (2011). DIVA 2.0: integrating daylight and thermal simulations using Rhinoceros 3D, Daysim and EnergyPlus. *Proc. Building Simulation 2011*. Ottawa: IBPSA.

Johan, R., & Wojciechowski, A. (2007). *A method for simulation based optimization using radial basis functions*. Master's Thesis. Göteborg University, Chalmers University of Technology, Department of Mathematics.

Jones, D.R., Schonlau, M., & Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization 13(4)*, 455–492.

Kicinger, R., Arciszewski, T., & Jong, K.D. (2005). Evolutionary computation and structural design: a survey of the state-of-the-art. *Computers & Structures 83(23–24)*, 1943–1978.

Kolarevic, B. (2005). Computing the performative. In *Performative Architecture Beyond Instrumentality* (Kolarevic, B., & Malkawi, A., Eds.), pp. 193–202. Hoboken, NJ: Taylor & Francis.

Kolarevic, B., & Malkawi, A. (2005). *Performative Architecture*. Hoboken, NJ: Taylor & Francis.

Kolda, T., Lewis, R., & Torczon, V. (2003). Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review 45(3)*, 385–482.

Koziel, S., Ciaurri, D.E., & Leifsson, L. (2011). Surrogate-based methods. In *Computational Optimization, Methods and Algorithms* (Koziel, S., & Yang, X.-S., Eds.), pp. 33–59. Heidelberg: Springer.

Lawson, B. (2004). *What Designers Know*. Oxford: Routledge.

Lawson, B. (2006). *How Designers Think: The Design Process Demystified*. Oxford: Elsevier/Architectural.

Lin, S.-H.E., & Gerber, D.J. (2014). Designing-in performance: a framework for evolutionary energy performance feedback in early stage design. *Automation in Construction 38*, 59–73.

Luebkeman, C., & Shea, K. (2005). CDO: computational design + optimization in building practice. *Arup Journal 3*, 17–21.

Malkawi, A. (2005). Performance simulation. In *Performative Architecture Beyond Instrumentality* (Kolarevic, B., & Malkawi, A., Eds.), pp. 85–95. Hoboken, NJ: Taylor & Francis.

Mardaljevic, J., Andersen, M., Roy, N., & Christoffersen, J. (2012). *Daylighting, Artificial Lighting and Non-Visual Effects Study for a Residential Building*, Velux Technical Report. Loughborough: Loughborough University, School of Civil and Building Engineering.

Miles, J. (2010). Genetic algorithms for design. In *Advances of Soft Computing in Engineering* (Waszczyszyn, Z., Ed.), pp. 1–56. Vienna: Springer.

Mullur, A.A., & Messac, A. (2005). Extended radial basis functions: more flexible and effective metamodeling. *AIAA Journal 43(6)*, 1306–1315.

Oxman, R. (2008). Performance-based design: current practices and research issues. *International Journal of Architectural Computing 6(1)*, 1–17.

Regis, R.G., & Shoemaker, C.A. (2007). A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing 19(4)*, 497–509.

Rios, L.M., & Sahinidis, N.V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization 56(3)*, 1247–1293.

Rittel, H.W., & Webber, M.M. (1973). Dilemmas in a general theory of planning. *Policy Sciences 4(2)*, 155–169.

Rutten, D. (2013). Galapagos: on the logic and limitations of generic solvers. *Architectural Design 83(2)*, 132–135.

Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards integrated performance-driven generative design tools. *Automation in Construction 14(2)*, 253–264.

Shi, X., & Yang, W. (2013). Performance-driven architectural design and optimization technique from a perspective of architects. *Automation in Construction 32*, 125–135.

Simpson, T.W., Poplinski, J.D., Koch, P.N., & Allen, J.K. (2001). Metamodels for computer-based engineering design: survey and recommendations. *Engineering With Computers 17(2)*, 129–150.

Smola, A.J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing 14(3)*, 199–1222.

Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization 11(4)*, 341–359.

Turner, C.J., Crawford, R.H., & Campbell, M.I. (2007). Global optimization of NURBs-based metamodels. *Engineering Optimization 39(3)*, 245–269.

Wang, G., & Shan, S. (2006). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design 129(4)*, 370–380.
Woodbury, R.F. (2010). *Elements of Parametric Design*. New York: Routledge.
Woodbury, R.F., & Burrow, A.L. (2006). Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 20(2)*, 63–82.
Yang, X.-S. (2010). *Engineering Optimization: An Introduction With Metaheuristic Applications*. Hoboken, NJ: Wiley.

Thomas Wortmann is a PhD student in the Architecture and Design Pillar at Singapore University of Technology and Design. He holds an MArch from the University of Kassel and a SMarchS in Design and Computation from MIT. He is an award-winning architectural designer and has been building computational design tools for over a decade. His research interest is computational design and optimization tools.

Alberto Costa is a Postdoctoral Researcher at Singapore University of Technology and Design. He holds BS and MS degrees in computer science from the University of Padova and a PhD in operations research from École Polytechnique, Palaiseau, France. Dr. Costa's research interests include blackbox optimization, reformulations in mathematical programming, clustering in networks, information retrieval, and recommender systems.

Giacomo Nannicini is an Assistant Professor in the Engineering Systems & Design Pillar at Singapore University of Technology and Design. He was previously a Research Fellow at Carnegie Mellon University Tepper School of Business and MIT Sloan. He has a PhD in computer science from École Polytechnique, Palaiseau, France. Dr. Nannicini's main research interest is combinatorial optimization, particularly linear and nonlinear integer programming, network optimization, dynamic programming, and (discrete) blackbox optimization.

Thomas Schroepfer is Associate Head of Pillar and Director of the Advanced Architecture Laboratory at Singapore University of Technology and Design. The laboratory explores the increasingly complex relationship between design and technology in architecture and urban design, relating to advances in environmental strategies, performance and energy, computer simulation and modeling, and building processes. His latest book, *Ecological Urban Architecture* (Basel: Birkhäuser, 2012), identifies strategies for achieving sustainable results on an urban scale. He is currently working with the Lee Kuan Yew Centre for Innovative Cities on a new book on dense and green building typologies, slated for publication in 2015.