DESIGN OPTIMIZATION
IN COMPUTER-AIDED ARCHITECTURAL DESIGN

RUCHI CHOUDHARY
College of Architecture, Georgia Institute of Technology,
Atlanta, GA 30332-0155, USA
Email address:ruchi.choudhary@coa.gatech.edu

AND

JEREMY MICHALEK
Department of Mechanical Engineering, University of Michigan,
Ann Arbor, USA 48109-2250, USA
Email address:michalek@umich.edu

Abstract. The proposition of using design optimization to formalize and add rigor to the decision-making process in building and construction was earlier compiled by Radford et al. in 1988, providing an in-depth demonstration of techniques available at the time. Much has changed since, both in the available solution methods and the nature of the problems themselves. This paper provides an updated insight into past and current trends of using this engineering design paradigm to solve architectural design problems, with an emphasis on continuous nonlinear formulations of simulation-based problems. The paper demonstrates different problem formulations and current techniques for solving them. Examples from recent research are used to demonstrate significant achievements and existing challenges associated with formalizing and solving decision-making tasks in architecture.

1. Introduction

Evaluating decisions through iterative interactions between design and analysis is common practice among building design and consulting teams. Rapid influxes of digital tools for representing and analyzing designs have offered unique opportunities to formalize and add rigor to the iterative decision-making process. As such, research and development efforts towards formal decision-support models have expanded along with automation in design, information technologies, and by availability of reliable analysis tools in form of simulation software. The distinctions between various formulations and methodologies that are available are rooted in different understandings and perceptions of the decision-making process. In conjunction, tools offering decision-support – ranging from search engines, integrated simulations, collaborative design, information exchange, and data management – have become the hallmarks of this current paradigm in computer-aided design research. In their various manifestations, the common objective emphasized by these efforts is to provide design decision-makers with information that facilitates their tasks – be it extensive catalogues of data-sets, explicit scoping of design alternatives, or meeting normative performance targets for benchmarking decisions.

By definition, the process of exploring design decisions that best match some specified criteria by subjecting decisions to repeated evaluations is embodied in the numerical design optimization model. In fact, the field of numerical optimization has been one of the primary influences in the rigorous formulation of normative decision-making models in engineering and management. It was also proposed as a decision-making model in building design by Wilson et al. in 1976, followed by a large (and still the most extensive) body of work by Gero et al. (1983) and Radford et al. (1988). Despite this decade of early work, the benefits of this formal decision-making model have remained largely untapped by the building research community. Ironically terms such as *optimal*, *optimum*, and *optimality* have become widely embedded in dialogues among design teams, but without any explicit definition of either the process or the end results they may represent. In the early 90's AI techniques came to be favored over numerical optimization models because of their logical rather than mathematical approach (Radford and Gero, 1988; Pohl, 1990; Shaviv, 1992; Malkawi; 1994). AI techniques are also less restrictive towards problems that are difficult to define mathematically and less demanding in terms of problem formulation. However, they have been shown to have limited applicability in practice.

In general, problems involving architectural design elements are often ill-defined: Many design elements are often selected by simultaneously considering a wide range of quantifiable as well as non-quantifiable criteria. Even when a problem allows numerical formulation, lack of explicit and standard evaluation criteria make the definition of design intents difficult. Few attempts at addressing this problem have resulted in building simulation software that offer ranking of some set building performance criteria with respect to predefined decision variables over and above the simulation. The main limitation of such models has been lack of flexibility in problem formulation and in the choice of analysis tools for function evaluations.

In addition to the nature of the problem, use of complex simulations to evaluate functions often yield undesirable properties in the optimization model, and therefore require special consideration of solution strategies that may be successfully used. Furthermore, the increased complexity of design tasks frequently requires multiple functions to be evaluated, many of which require different analysis tools. The last two points have also challenged engineering design problems, resulting in vast improvements in numerical methods and solution strategies for posing and solving large-scale simulation based design tasks. In view of recent developments we present an updated insight into design optimization models with citations of recent work that address past challenges and demonstrate benefits that may be claimed towards an explicit and rigorous decision-making process.

In the following sections we review the general mathematical formulation of the nonlinear programming optimization model and highlight some distinguishing properties that classify models and solution methods. This review largely includes common solution methods that have been applied to design problems in architecture with an emphasis on continuous nonlinear formulations for simulation-based problems.

2. The Design Optimization Model

A general multicriteria optimization problem for optimizing a design involves (1) parameterizing design alternatives using a vector of design variables **x**, which can be manipulated by the designer to alter the design, (2) defining constraint functions **g**(**x**) and **h**(**x**) that specify the range of values for **x** that correspond to feasible and meaningful designs, and (3) defining an objective function $f(\mathbf{x})$ or vector of objective functions **f**(**x**) that describe the goal or goals to be attained by the design. Each of these functions may involve simple relationships and/or complex simulations. Formally, the mathematical statement of this design problem is posed as:

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le 0 \quad \mathbf{h}(\mathbf{x}) = 0 \\ &\qquad\qquad\quad \mathbf{x} \in \aleph \subseteq \Re^n \end{aligned} \tag{1}$$

where $\aleph$ is the set constraint of the n-dimensional real space $\Re^n$ (Papalambros and Wilde, 2000). Once the problem is formulated mathematically with a set of variables, objectives and constraints, general optimization algorithms can be applied to find the optimal solution(s). In this iterative search for optimal solutions, building a good optimization model and choosing appropriate methods to solve the problem are crucial for finding meaningful solutions that meet stated goals in an efficient manner.

Careful model building is a prerequisite in optimization. A fair portion of this task depends on the modeler's understanding of the problem and model properties that will affect the solution process. Building an optimization model involves translating a problem statement into a mathematical model and requires a thorough understanding of the different forms equation (1) can take. Typically the mathematical model results from the nature of the problem. Often though, this model is 'designed,' keeping in mind that its form will determine the solution methods that can be applied to solve it, the optimization process, and the quality of results.

Since the use of optimization models in simulation-based building design problems has been so limited (structural design being an exception), there is no comprehensive source for educating the user on nuances of model formulation specific to architectural problems since Radford and Gero (1988). In general, optimization models can be characterized based on the mathematical properties of variables and functions, such as continuous, discrete, or mixed-discrete. Often, a particular situation can be modeled in more than one way. For instance, the length of a room can be either represented as a continuous range or incremented over discrete lengths (for example, in 1 meter intervals) depending on its sensitivity to the objective function. This broad model classification is often used to guide model building and select appropriate solution strategies.

3. Solution Methods

Methods used for solving optimization problems can be distinguished as gradient-based or derivative-free. While implementations of most solution methods are now available as software packages, it is important to understand how these methods work to be able to select a suitable method for a particular problem, to apply the method appropriately, and to troubleshoot when necessary. The following sections provide an overview of some of these methods with examples of their use for solving simulation-based design problems in architecture.

3.1. GRADIENT-BASED METHODS

Gradient-based methods are proven to converge to local minima for formulations with continuous and smooth functions with respect to all decision variables. So a general assumption for continuous problems is that the functions are also differentiable with respect to decision variables. However, gradient-based methods are commonly used to solve any problem with variables defined over a continuous domain. Gradient-based methods iteratively move from one design alternative to another improved alternative, until no better solution can be found in the surrounding neighborhood of the design space. They are called “gradient-based” because they use information about the gradient of the objective and constraint functions to guide the direction of search while looking for improved alternative designs. Common examples under this typology of solution methods include generalized reduced gradient (GRG) and sequential quadratic programming (SQP).

Simulation-based building performance functions are often continuous functions in theory, and where possible, gradient-based methods have been shown to be very efficient and reliable for successfully solving problems ranging from shape and partitions in buildings (Jedrzejuk et al., 2002), layout design (Michalek et al., 2002), and computing optimal control strategies for time-scheduled operation in buildings (Zaheer-uddin et al., 2000) or specific design elements (Park et al., 2003). They are typically used because they are fast, rigorous, guarantee a locally optimum solution, and can handle large numbers of variables and constraints. If the problem is uni-modal, then the local optimum is also the global optimum. However architectural design problems constituting multiple criteria are often multi-modal (*i.e.* there are multiple local minima – each of which is better than all nearby designs, but not necessarily better than all alternatives). For multi-modal problems, a *multistart* strategy can be applied, which means that the optimization is run repeatedly with different starting points (initial values of decision variables), and the lowest of the local minima found among the results is taken to be a good local optimum or possibly the global optimum. Generally the modeler will have intuition about the design problem and can choose starting points to find local minima in the area of interest. Some general purpose gradient-based solvers include multistart strategies within their implementation to guide this process.

Other common problems using gradient-based methods result from undesirable mathematical properties of the optimization model such as non-smoothness, or discontinuities. Lack of proper *scaling* can also lead to numerical difficulties and usually occurs when variables or model functions have values of different orders of magnitude. Problems with scaling can be generally avoided by normalizing all variables and functions. However, with complex simulations it is often difficult to guarantee mathematical properties of model functions. In fact, use of complex simulations often results in derivative discontinuities that can cause gradient-based methods to fail even when the problem is continuous. For example, function responses from airflow models based on computational fluid dynamics or heat balance models based on finite element analysis are often 'noisy'. Such cases require special considerations such as selecting methods that do not rely on directional or derivative information of functions.

3.2. DERIVATIVE FREE METHODS

Derivative free methods do not rely on function gradients to guide the search process or evaluate the optimum solution. Because derivative-free methods do not use gradient information, they are particularly attractive for discrete problems, for problems that have discontinuous or noisy function responses, or for problems that have disconnected feasible domains. Furthermore, most derivative free methods search the design space globally and are therefore also well suited for multi-modal problems.

Methods within this category work through repeated function evaluations, accepting and rejecting candidate solutions, and the search for an optimum proceeds iteratively using heuristics, which can vary from 'highly intelligent' search techniques to simple random search. In some methods local search is also included by enlarging and searching the area around a successful iteration. However, due to lack of gradient information to guide search the algorithms can take a long time to converge to a local minimum around a good candidate solution, especially if the problem is large.

Unlike gradient-based methods, there is no canonical optimality test to define convergence or check for optimality of a solution. Instead, convergence properties are specific to the solution method. So while some methods may have proven convergence to a global optimum, others yield the solution found on stopping at specified termination criteria. This broad spectrum of derivative-free methods can be classified into *deterministic* and *stochastic* methods. Derivative free deterministic methods will always produce the same solution given the same input. Contrarily, stochastic methods contain a random element in the search process and hence may or may not arrive at the same solution when the problem is solved multiple times.

The main advantage of derivative-free deterministic methods is that they are repeatable without requiring restrictive mathematical properties from model functions. Also, they incorporate both global and local search, and if run long enough, will find the global optimum, at least in the limit. Derivative-free deterministic methods such as generalized pattern search (GPS), divided rectangles (DIRECT), and lattice methods have been shown to perform particularly well. Saporito et al. (2001) have demonstrated the use of Lattice Methods for studying the combined effects of multiple variables for the goal of minimizing energy use in lieu of sensitivity analysis techniques. Function evaluations in their case are simulation-based, and they were able to obtain optimum values of energy consumption when the number of variables was reduced to a small set. Peippo et al. (1999) demonstrate a model that uses pattern search to find the economic optimum with respect to several physical and technical features of buildings. Their work shows how mixed-discrete problems can be implemented easily with pattern search algorithms. However, they also report slow convergence, with 50 variables requiring up to 10000 simulation calls. A major drawback is that methods in this category have difficulty solving problems with more than a very modest number of variables (over 10 variables). In addition, if the functions are expensive in terms of computational run-time, the total time required to find a solution can be prohibitive. Another disadvantage of these methods is that they do not necessarily converge quickly to the minimum. The algorithm may get close to the minimum, but because it proceeds with lack of valuable derivative information, it often takes very long to locate the exact minimum.

Like deterministic methods, stochastic methods such as simulated annealing (SA) and genetic algorithms (GA) are also very versatile and as a result applicable to a wide range of problems. Both are conceptually inspired by natural phenomenon (simulated annealing from thermodynamics, and genetic algorithms from evolution) and based on drawing analogies between "growth towards improvement" in nature and improvement in a current design state.

Over the past few years stochastic methods have been applied to a range of mixed-discrete problems for optimizing thermal and lighting performance based on building enclosure, HVAC design, and control schedules (Wright et al., 2002; Choudhary and Malkawi, 2002; Caldas et al., 2002; Coley et al., 2002). The success of these methods rely to a large extent on how the algorithm is setup; and setting all the parameters can be quite arduous and require considerable intuition and experience. Also, the convergence rate of these methods is generally slow. Choudhary et al. (2002) report results from a mixed-discrete constrained problem for optimizing ventilation efficiency using GA with respect to duct layout and room geometry. They demonstrate that it is hard to gauge *a priori* how long the algorithm should be run to be able to arrive at a "good enough" solution – especially when the runtime is even longer because the problem is simulation-based.

Another important feature of most derivative free methods is that they do not handle constraints explicitly. Choudhary et al. (2002) formulate constraints by formulating an auxiliary objective function, which includes penalties for constraint violations added to the objective function – which is the most common and general strategy. Another way of ensuring constraint satisfaction is to restrict the search, which means defining the set of possible search moves such that infeasible points are impossible. This is a problem-dependent approach, and more details can be found in the literature on heuristic search techniques (for example, Michalewicz and Fogel, 2000; Reeves, 1993).

Recent applications (Whetter and Polak, 2003; Choudhary 2004) have addressed issues of large computational runtime associated with derivative-free methods by using approximation-based methods that derive simpler functions of the original simulation responses and use them for a partial search during the optimization process. For instance, Choudhary (2004) used an approximation based technique called Efficient Global Optimization (EGO), (Jones et al., 1998, and later extended by Sasena, 2002) that fits a response surface to an initial data sample of the objective function and applies the DIRECT algorithm to simultaneously search for the minimum on the response surface and improve the response surface by sampling more points in areas of high uncertainty. The work showed significant reductions in total run-time when compared to using the DIRECT algorithm. Such techniques have earlier been demonstrated in solving many engineering design problems.

3.3. HYBRID METHODS

Hybrid strategies combining two or more methods are sometimes used to overcome problems associated with one particular method. For example, Michalek et al. (2002) demonstrate a hybrid strategy combining two solution algorithms for finding the global optimum for an architectural layout design problem involving multiple objectives of cost, function, and aesthetics. Their model offers a new approach to solving multi-modal and mixed-discrete optimization problems that takes advantage of the efficiency of gradient-based algorithms (SQP), where appropriate, and uses evolutionary algorithms (SA) to make discrete decisions and do global search. Monks et al. (2000) have also demonstrated a similar approach for solving acoustic design optimization problems. Wetter and Wright (2003) propose to combine GA and pattern search to derive a hybrid method that reduces computational run time in problems involving expensive simulations. Such strategies have been shown to be successful where using one or another method for solving problems may fail.

4. Large-scale Optimization

The selection of a suitable method depends on the problem formulation and the goals of performing the optimization. Often though, we encounter design problems that are not only complex, simulation-based formulations, but are also large – requiring evaluation of multiple functions that use diverse simulations. In principle it is desirable to evaluate interrelated design decisions concurrently so that their combined effects on different functions may be maintained. However, combining all design decisions and evaluating them simultaneously is often difficult because it involves multiple and often conflicting functions that may require expert analysis at very different levels of complexity and with different kinds of design information. Past work on the simultaneous optimization of a design to achieve multiple criteria has used multi-criteria formulations with preference or non-preference based strategies. They will typically provide the decision maker with values for decision variables that best accommodate a weighted set of performance criteria. The difficulty in elaborating these efforts to include more than a few analytical tasks and decision variables is that the problem quickly becomes too large and complex to be implemented in one model. Even when numerical results are successfully obtained, one may not be able to interpret the design trade-offs or use intuition to confirm computed results (Papalambros 2001), and the high-dimensionality of some problems make solutions difficult or impossible to find in practice. For large and complex cases, some form of problem decomposition therefore becomes necessary. In addition to making a problem manageable, decomposition of a large problem by focus or discipline is beneficial because it allows the specialized analysis and decision-making of individual design tasks. On the other hand, when the decision-maker separates and designs individual parts of the problem, he must not only coordinate common decisions between different problems, but also combine the solutions into a single compatible set.

In a recent work, Choudhary (2004) shows that it is possible to treat large scale architectural problems as systems design problem by extending a hierarchical design optimization model called analytic target cascading (ATC) to simulation-based design contexts in architecture. ATC is a hierarchical optimization methodology for achieving compatible design targets in large engineering systems at early product development stages (Kim, 2001; Kim et al., 2001). The following main assumptions apply for extending ATC as a methodology for handling simulation-based design problems in architecture: (a) a complex simulation-based design problem can be decomposed or partitioned into subproblems; (b) it is possible to identify a hierarchical organization in the decomposition; and (c) building performance goals can be embodied as targets that are to be achieved via design decisions, and some of the performance goals can be set and introduced as part of initial problem definition. This work includes illustrative case studies involving optimization of HVAC capacity, energy consumption, thermal comfort, and ventilation efficiency targets for complex and critical problems (where meeting prescribed values of some performance targets is highly prioritized). The decision variables included in the model are decomposed hierarchically into subproblems and a coordination strategy is applied to iterate through the multi-level structure.

In the context of simulation-based design, a particularly beneficial feature of this decomposition-coordination approach is that each subproblem in the hierarchy constitutes a separate optimization problem and is associated to only those analysis models that are capable for computing the values of performance goals set for it. This allows both optimization algorithms and analysis tools to be used exclusively for the relevant decision-making subproblem, while coordinating all such subproblems to achieve a consistent solution that is optimal with respect to the overall system.

5. Concluding Remarks

Properties of the optimization model will typically have a significant bearing on which method can be applied to solve a problem or a subproblem. For instance, gradient-based methods are fast and rigorous for solving continuous problems, while a large combinatorial problem can be best handled using stochastic methods. The quality of the final solution is another aspect influenced by the optimization algorithm and solution strategy. Gradient-based methods search locally, but if the problem is well understood and if the process can be started from a ‘point of interest’, then a local minimum will suffice. Otherwise, multiple starting points will often derive the global optimum. Derivative-free require a large number of function evaluations to converge which is almost always impractical for simulation-based problems. So here again it is usually good to have a vision of what would suffice as a ‘good enough’ solution. This compromise between quality of the solution and computational time almost always exist. Sometimes, two methods can be used in conjunction with each other and such hybrid strategies often perform very well for combining both global and local search. Rigor and mathematical precision are generally preferred qualities for a process, but they are not always possible because of the nature of problems architectural designers want to solve and for the technical skills they demand. So at best, the most suitable method is one which is well understood so that it can be used efficiently to yield desired solutions and to enhance the knowledge of the problem itself.

In addition to exploring a range of different methods, the applications from the past few years also demonstrate that with good understanding of the methods involved, design optimization can be effectively used to improve building performance and provide rigor in the way we use simulation tools. In a larger perspective, this body of work revealed several areas and subsequent questions that require further research. For example, establishing standard building evaluation functions (such as performance indicators) along with their sensitivity to typical decision variables can help the user immensely in formulating appropriate objectives functions. Recent work called the Design Analysis Integration (DAI) (Augenbroe et al., 2003) has addressed this by defining and correlating specific design problems to their corresponding analysis functions.

From an implementation perspective the main requirement is of prototypes that are flexible and can yet assist the user in problem formulation and in organization of decision-making tasks by rigorous criteria. This essentially implies a general framework that contains a repository of decision and analysis models, assists the modeler in mathematical formulation of the decision models, and incorporates organizational aspects such as book-keeping – a task that would require a pool of effort from different sub-fields of computer-aided architectural design.

Acknowledgements

This paper is dedicated to the Optimal Design Laboratory at University of Michigan.

References

Augenbroe, G., et al.: 2003, The Design Analysis Integration (DAI) Initiative, *Proceedings of the Eighth International IBPSA Conference*, Eindhoven, The Netherlands.

Caldas L. G., and L. K. Norford: 2002, A Design Optimization Tool Based on a Genetic Algorithm, *Automation in Construction*, 11, 173-184.

Choudhary, R., and A. Malkawi: 2002, Integration of CFD and Genetic Algorithms, *Proceedings of the Eighth International Conference on Air Distribution in Rooms*. Copenhagen, Denmark.

Choudhary R.: 2004, *A Hierarchical Design Optimization Framework for Simulation-based Architectural Design*, Ph. D. diss., University of Michigan, USA.

Coley, D. A., and S. Schukat: 2002, Low-energy Design: combining computer-based optimization and human judgment, *Building and Environment*, 37(12), 1241-1247.

Gero J. S., Neville D'C., and A. D. Radford: 1983, Energy in Context: A multicriteria model for building design, *Building and Environment*, 18(3), 99-107.

Jedrzejuk, H., and W. Marks: 2002, Optimization of Shape and Functional Structure of Buildings as well as Heat Source Utilization Example, *Building and Environment*, 37(12), 1249-1253.

Kim, H. M., et al.: 2001, Analytical Target Cascading in Automotive Vehicle Design, *Proceedings of the 2001 ASME Design Automation Conference*, Pittsburgh, Pennsylvania.

Kim, H. M.: 2001, *Target Cascading in Optimal System Design*, Ph. D. diss., University of Michigan, USA.

Malkawi A.: 1994, *Building Energy Design and Optimization: Intelligent Computer-aided Thermal Design*, Ph. D. diss., Georgia Institute of Technology, USA.

Michalek, J., R. Choudhary, and P. Y. Papalambros: 2002, Architectural Layout Design Optimization, *Engineering Optimization*, 34(5), 461-484.

Michalewicz, Z., and D. B. Fogel: 2000, *How to Solve It: Modern Heuristics*, Springer-Verlag, Berlin.

Monks, M., B. M. Oh, and D. Julie: 2000, Audioptimization: Goal-Based Acoustic Design, *IEEE Journal*, 0272-1716, 76-91.

Papalambros, P.Y.: 2001, Analytical Target Cascading in Product Development, *Proceedings of the 3rd ASMO UK / ISSMO Conference on Engineering Design Optimization*, Harrogate, North Yorkshire, England.

Papalambros P. Y., and D. Wilde: 2000, *Principles of Optimal Design: modeling and computation.* 2nd ed., Cambridge University Press, New York.

Park, C., G. Augenbroe, and T. Messadi: 2003, Daylighting Optimization in Smart Façade Systems, *Proceedings of the Eighth International IBPSA Conference*, The Netherlands.

Peippo, K., P. D. Lund, and E. Vartiainen: 1999, Multivariate Optimization of Design Trade-offs for Solar Low Energy Buildings, *Energy and Buildings*, 29(2), 189-205.

Pohl J., Myers, L., and A. Chapman, ICADS: 1990, An Intelligent Computer-aided Design Environment, *ASHRAE Transactions*, 96(2), 473-480.

Radford, A. D., and J. S. Gero: 1988, *Design by Optimization in Architecture and Building*, Van Nostrand Reinhold, New York.

Reeves, C. R., ed.: 1993, *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, London.

Saporito A., et al.: 2001, Multi-Parameter Building Thermal Analysis Using the Lattice Method for Global Optimization, *Energy and Buildings*, 33(3), 267-274.

Sasena, M. J.: 2002, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph. D. diss., University of Michigan, USA.

Shaviv, E., and Y. E. Kalay: 1992, Combined Procedural and Heuristic Method to Energy-conscious Building Design and Evaluation, *in* Y. E. Kalay(ed), *Evaluating and Predicting Design Performance*, John Wiley and Sons, New York.

Wetter, M., and E. Polak: 2003, A Convergent Optimization Method Using Pattern Search Algorithms with Adaptive Precision Simulation, *Proceedings of the Eighth International IBPSA Conference*, Eindhoven, Netherlands.

Wetter, M., and J. Wright: 2003, Comparison of a Generalized Pattern Search and a Genetic Algorithm Optimization Method, *Proceedings of the Eighth International IBPSA Conference*, Eindhoven, Netherlands.

Wilson A. J., and A. B. Templeman: 1976, An Approach to the Optimum Thermal Design of Office Buildings, *Building and Environment*, 11(1), 39-50.

Wright, J. A., H. A. Loosemore, and R. Farmani: 2002, Optimization of Building Thermal Design and Control by Multi-criterion Genetic Algorithm, *Energy and Buildings*, 34(9) 959-972.

Zaheer-uddin, M., and G. R. Zheng: 2000, Optimal Control of Time-scheduled Heating, Ventilating and Air Conditioning Processes in Buildings, *Energy Conversion and Management*, 41(1), 49-60.