Title:
Integrating Grasshopper and Matlab for Shape Optimization and Structural Form-Finding of Buildings

Authors:
Nixon Wonoto, nwonoto@clemson.edu, Clemson University, USA
Vincent Blouin, vblouin@clemson.edu, Clemson University, USA

Keywords:
Structural Optimization, Grasshopper, Finite Element Analysis, Genetic Algorithm, Form-Finding.

DOI: 10.14733/cadconfP.2018.288-292

Introduction:
There has been an increasing number of architecture publications that demonstrate the potential of structural optimization as a form-finding technique in the architectural schematic design phase of buildings. However, most structural optimization research case studies tend to reduce practical design and analysis problems into simplified theoretical models in which materiality, geometry and loading conditions are over-simplified. We suspect that the need for simplification is due to the difficulty of programming complex processes with the commonly used software.

This paper presents a structural optimization case study that strives to allow the inclusion of complexity. The workflow integrates the architectural algorithmic modeling tool, Grasshopper, and the high-performance computing language, Matlab. Grasshopper, a visual programming language in Rhino, is used as the generator of the geometry and the platform that allows convenient structural parametrization. The optimization process includes an automated update of structural size, shape, and topology, material properties, and loading conditions as functions of the design variables. A versatile finite element analysis and genetic algorithm code are programmed in Matlab to carry out the mixed nonlinear and integer programming optimization process. The method is applied to a parametric skyscraper design problem to demonstrate how the use of Grasshopper can expedite the implementation of a complex problem and thereby facilitate the architectural schematic design phase. An exhaustive analysis of the design space was also carried out to validate the results of the method.

Main Idea:
Structural optimization and form-finding use structural performance as the primary driver for selecting optimal candidates from the design space. Structural optimization was often exclusive to the engineering field but has recently swayed into the field of architecture. In line with the emergence of design computation in architecture, there have been burgeoning research reports in architecture focused on the idea of an automated form-finding process for designing large concrete roofs [5], domes [3], Voronoi's cell structures [2], trusses [1], tessellated structures [6] and Miura origami fold retractable roofs [7].

In spite of these advancements, there has not been many architectural structural optimization frameworks that incorporate methods that allow easy consideration of the complexity associated with realistic design problems. More specifically, design difficulties are related to the dependency on design variables of the loading conditions, material properties, size, shape and topology properties, and the utilization of multiple structural types. Thus, real design and analysis problems are usually reduced to over-simplified exercices.

Even though one may argue that the architectural schematic design phase does not require detailed structural definition, the authors believe that over-simplification is partly responsible for the renowned difficulties of the collaboration between architects and engineers during the building design process. The premise of this paper is to overcome these issues by encouraging the use of semi-automated CAD processes that capture realistic structural behaviors to reduce the programming burden. The paper presents a versatile method for efficiently carrying out architectural structural optimization that includes the complexity of a typical engineering problem in designing complex structures. In this problem, an analytic form of the objective function is not known but is evaluated through performance analysis. Such a problem is classified as a black-box optimization problem, and simulation-based optimization [4]. The method integrates Grasshopper and Matlab for carrying the structural optimization procedure.

Method

The integration between Grasshopper, the customized Matlab finite element analysis (FEA), and a customized genetic algorithm (GA) optimizer is carried out in an iterative manner as shown in Figure 1. Grasshopper is used to generate a parametric model. It then manages and updates the structural analysis FEA input file based on the values of the design variables passed by Matlab during the optimization process. The output from Grasshopper includes nodal coordinates, element connectivity, and all information related to the structural analysis, including the loading conditions, boundary conditions, material properties, section property, element types, and type of analysis. The advantage of managing the structural analysis setup in Grasshopper is to have full control over the analysis configuration. Instead of predetermining the values of the analysis configuration, the method adjusts the analysis format automatically depending on the given permutation of the geometry.

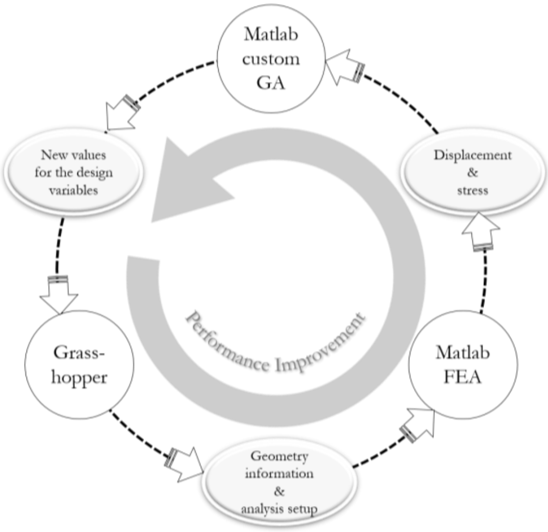

Fig. 1: Workflow of the presented structural optimization method.

Case Study

The developed structural optimization method is used for designing a twisted skyscraper, under both dead and wind loads. The twist of the building is assumed to be the most important architectural feature. The case was designed such that the skyscraper employs the exterior diagrid for the braced tubular structural system, using the combination of frames and trusses at the outside perimeter as the primary structure. Four design variables are used including the radius of the floor located in the mid-section of the skyscraper (s1), the radius of the top floor of the building (s2), the twist angle of the tower in radians (s3), and the number of floors (s4), which controls the overall height of the building since the height of each floor is assumed to be constant.

When the building is twisted, each node in the structure carries a unique magnitude of the load. The value of each wind pressure coefficient, noted cp, corresponding to each node of the structure at a given floor is a function of the twist angle. The value of cp is divided into x and y components and tabulated into matrices that are updated at each iteration of the optimization process to give the proper loading condition corresponding to the particular design permutation. Figure 2 shows cp values at a sample permutation of the skyscraper. There is a total of 24 sections corresponding to four bracing segments at each edge of the hexagonal floor plan.

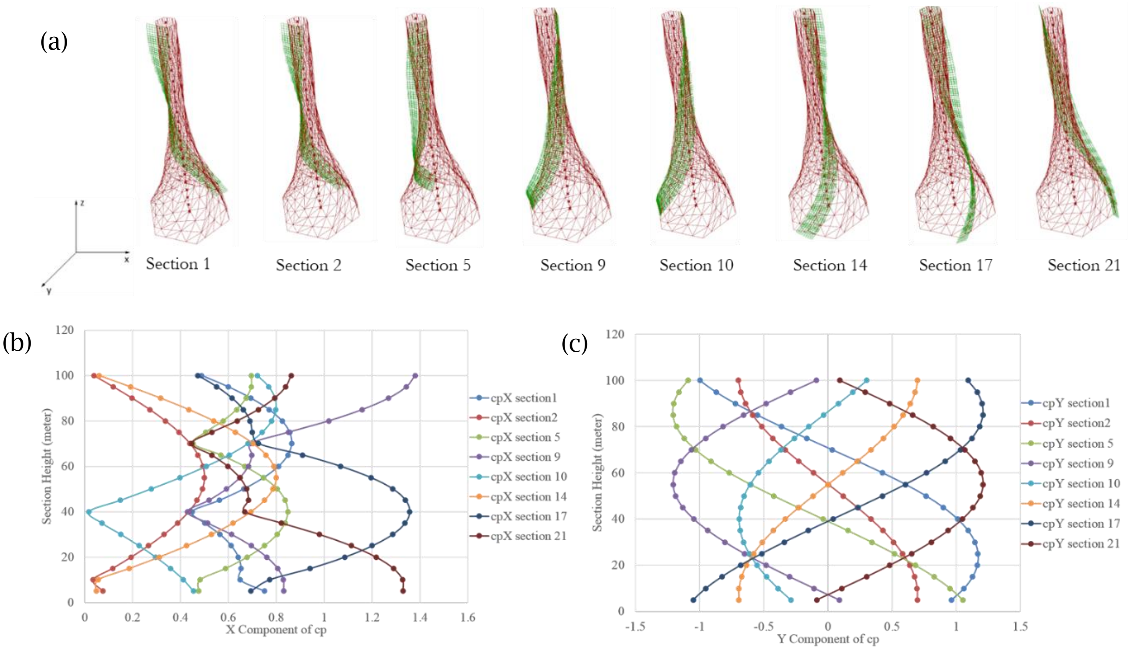

Fig. 2: (a) Sampled sections of the structures at a sampled design permutation, (b) X component of cp vs. the sampled sections, and (c) Y component of cp vs. the sampled sections.

The performance of the genetic algorithm generally depends on the population size and the total number of generations for carrying out the structural optimization. The results reported in this paper were obtained using a population size of fifty individuals and forty generations. The computational time of the analysis of each individual of each iteration of the optimization is mostly the execution time of Grasshopper and the FEA. In the particular case of the skyscraper design, Grasshopper and the FEA require an average computational time of 46 seconds and 53 seconds, respectively. As shown in Figure 3, the maximum fitness value, which corresponds to maximizing the twist angle of the building increases drastically from the first to the third generation. The further slight increase occurs in the 20^{th} generation. The trend line of the average fitness values in each generation also increases. The fluctuation in the average fitness values shows that the stochastic process involving the crossover and mutation in the algorithm perform as desired.

An exhaustive search analysis of the design space was then performed to validate the results found by the genetic algorithm. The design space is discretized using 10 levels for s1 and s2 and 8 levels for s3 and s4. The full factorial analysis of the design space corresponds to a total of 6400 analyses. The overall computational time of the exhaustive search is approximately ten times greater than that of the genetic algorithm.

Comparing the results of the genetic algorithm and the exhaustive search analysis, the optimization problem seems to be multimodal. The global optimum was shown to have a twist of 2.56 rad and twenty-one floors. Despite not achieving the global optimum, the presented method proves to be successful to find the improved candidates in a much shorter time compared to the exhaustive search analysis, and thus is a viable form-finding technique for dealing with a complex mixed integer nonlinear optimization problem during the architectural design process.

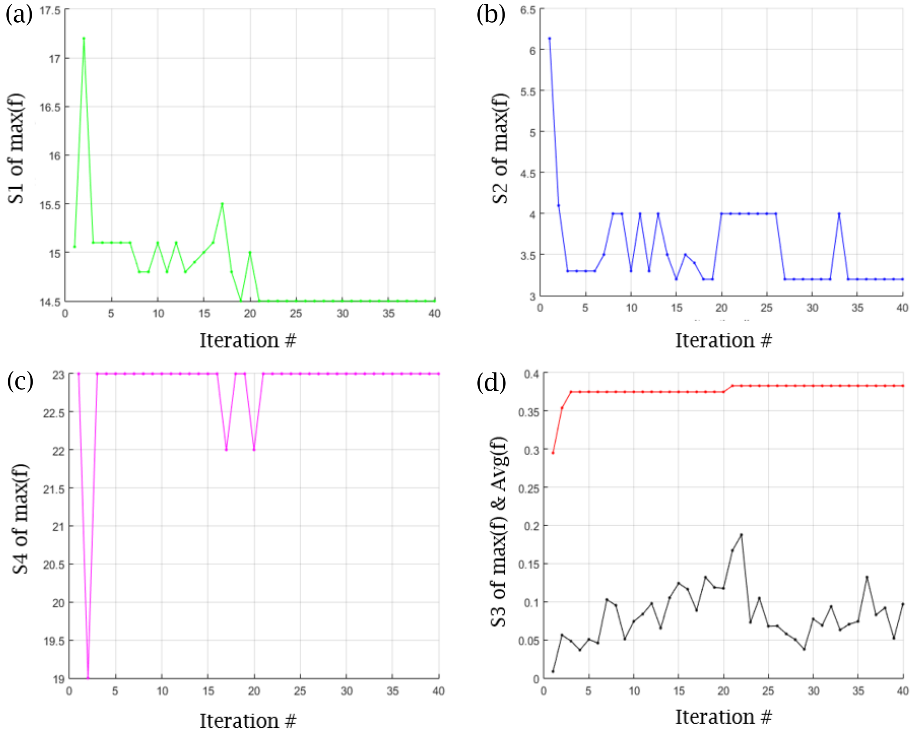

Fig. 3: (a) Evolution of s1, (b) s2, (c) s4, (d) s3 (red) and average fitness (black) as functions of number of iterations.

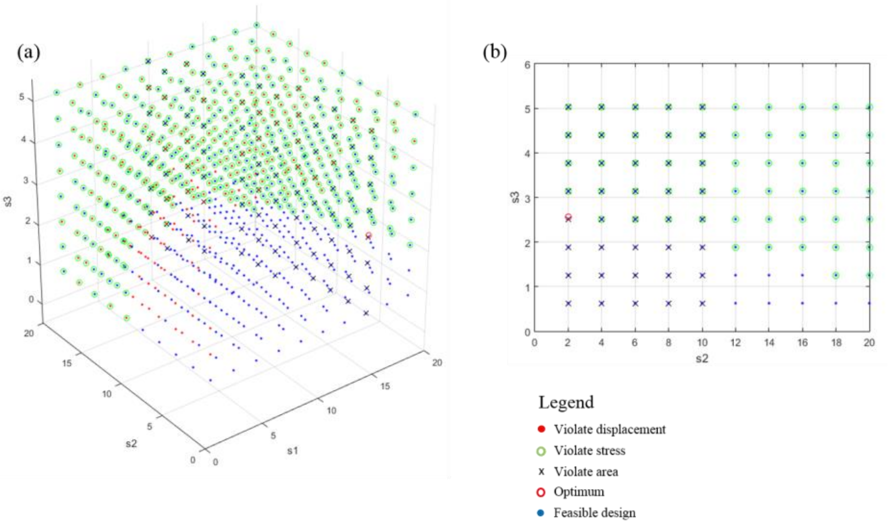

Fig. 4: Graphical representation of the design space showing feasible designs and the optimum using the exhaustive search analysis: (a) s1 vs. s2 vs. s3 at s4=21; (b) s2 vs. s3 at s1=16 and s4=21.

Conclusion:

This paper presents a form-finding architectural, structural optimization method that couples the architectural visual programming language, Grasshopper, and a high-performance computing language, Matlab. An instance of a parametric model of a twisted skyscraper was developed to demonstrate the capability of the method to carry out a complex mixed integer nonlinear programming problem. The presented method can conveniently deal with problems that demand high complexity in the geometry configuration. By using the presented workflow, the wind loads, dead loads, section properties, building's floor areas, and the number of structural connections can be taken as functions and automatically updated by Grasshopper during the optimization process.

References:

[1] Felkner, J.; Chatzi, E.; Kotnik, T.: Architectural feedback in the structural optimization process, Structures and Architecture: Concepts, Applications, and Challenges, 2013.
[2] Friedrich, E.: The Voronoi diagram in structural optimization. M.S. Thesis, Bartlett School of Graduate Studies, 2008.
[3] Haddad, T.: Integrating structural feedback into design processes for complex surface-active form. M.Arch. Thesis, Georgia Institute of Technology, 2006.
[4] Ky, V.; Ambrosio, C.; Hamadi, Y.; Liberti, L.: Surrogate-based methods for black-box optimization. International Transactions in Operational Research 24(3), 2015, 393-424.
[5] Sassone, M.; Pugnale, A.: Evolutionary structural optimization in shell designs, Advanced Numerical Analysis of Shell-like Structures, 2007.
[6] Von Buelow, P.; Falk, A.; Turrin, M.: Optimization of the structural form using a genetic algorithm to search associative parametric geometry, Proceedings of the International Conference on Structures Architecture, 2010. https://doi.org/10.1201/b10428-93
[7] Wonoto, N.; Baerlecken, D.; Gentry, R.; Swarts, M.: Parametric design and structural analysis of deployable origami tessellation. Computer-Aided Design and Applications, 10(6), 2013, 939-951. https://doi.org/10.3722/cadaps.2013.939-951

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 288-292