ABSTRACT

In this study, we introduce a new approach that incorporates form diversity into architectural design optimization, which will potentially accommodate designers’ aesthetic judgment into the whole building optimization process. Form diversity is defined here as the level of difference in building geometric forms. We developed a form comparison algorithm to lead to a reasonable number of optimal design solutions of highly diverse forms. This allows for a post-optimization articulation of preferred solutions, and helps satisfy the aesthetic criterion in parallel to the measurable objectives.

The methodology involves experimenting and prototyping. Experiments were done at different progress levels of the optimization tasks to test the feasibility of the system’s framework. A prototype framework was developed using parametric modeling, energy simulation, daylight simulation, Pareto optimization, and Multi-Objective Genetic Algorithms. The initial results demonstrate that the system has the capability to successfully work as desired with possible improvements. Comparison of results before and after shape comparison is discussed.
INTRODUCTION
Architectural design can be considered as a multi-objective optimization (MOO) problem in which multiple potentially conflicting criteria are sought to be simultaneously satisfied (Gero 2012; Radford and Gero 1987). However, the MOO approaches have been primarily applied to only quantifiable (numerically expressible) objectives related to a building’s functional performance (Yan et al. 2015), often leading to aesthetically unpleasing design solutions (Brown et al. 2015). Form aesthetics is considered one of the most important architectural design intents in the building design process (LaHood and Brink 2009).

Despite the fact that aesthetics are still beyond the capability of current computer-aided design systems (Galanter 2012), the aesthetic criterion is always embedded in the designers’ expertise and appears in their implicit use of aesthetic judgment (Reich 1993). However, implicit consideration of aesthetics does not sufficiently solve the issue of aesthetics’ inclusion in architectural design. Two associated problems were identified in current architectural MOO models. First, there is a lack of methods that explicitly insert the designers’ aesthetic preferences into optimization. The second issue is the need for new methods to decrease the often overwhelming number of final solutions in the Pareto front (e.g. more than a hundred of final solutions) to a manageable number of diverse solutions in order to facilitate decision making (Yang and Wang 2012).

In response, this work investigates a new interactive architectural design optimization system that can potentially help accommodate designers’ aesthetic judgment into architectural design optimization, by allowing their choices to include desired, or discard undesired, design solutions in the MOO process. We introduce a form diversity algorithm to lead to a reasonable number of design solutions of highly diverse forms. Form diversity is defined here as the level of difference in building geometric forms. This requires developing an algorithm that compares building forms, eliminates similar shapes, and ensures form diversity in design solutions. This post-optimization articulation of preferred solutions helps satisfy the aesthetic criterion in parallel to more measurable objectives.

BACKGROUND
This work incorporates form diversity into the optimization framework for achieving potential non-quantifiable objectives, and creates and tests a working prototype for the suggested method. The prototype utilizes MOO, genetic algorithms, parametric modeling, energy simulation and daylight analysis, and a designer’s aesthetic judgment and form diversity.

Multi-Objective Optimization (MOO)
There are two approaches to solve MOO problems: 1) combine the multiple objectives into a single (composite) objective by determining a weight for each objective and using the weighted sum method, and 2) find a Pareto optimal solution set, which is a set of optimal solutions that are not dominated by each other (Konak et al. 2006). When a change to a parameter value of a solution leads to an improvement in one objective without making any other objective worse off, it is considered a Pareto improvement; when no Pareto improvements could be made, the solutions are non-inferior or non-dominated and called Pareto optimal (Radford and Gero 1987). This research uses a Pareto optimality method in MOO.

Genetic Algorithms (GAs) for MOO
Genetic Algorithms (GAs) are heuristic and stochastic search algorithms, mainly applied for solving optimization problems with defined objectives and parameters (Yi et al. 2012). Being a population-based approach, GAs are particularly customizable to accommodate MOO problems by using specialized fitness functions and the possibility of introducing solution diversity (Konak et al. 2006). A multi-objective genetic algorithm (MOGA) is based on Pareto analysis that assists the algorithm in the optimization of all its objectives. This work utilizes MOGA in the prototyping process, though other optimization algorithms may be utilized as well.

Parametric Modeling
Parametric modeling is the use of geometry modeling when rules are applied to govern geometric (or other) relations within design morphing for modification (Holzer 2015). Parametric modeling allows for variation in design using variables and constraints in model elements and their relations, and enables generative explorations of design options that automatically update and change according to updated performance requirements (Aish and Woodbury 2005). The use of parametric modeling is important for this research to allow for changes of building parameters to occur and lead to variations of design solutions to satisfy design objectives.

Energy Simulation and Daylight Analysis
Building simulation has emerged through energy reduction demands and sustainable practices (Clarke 2001). Simulation of daylighting performance is also related to the recent escalated focus on energy efficiency and environmentally conscious building design (Elghazi et al. 2014; Lagios et al. 2010). Energy footprint and daylight measure are targeted in this study as the quantifiable performance objectives to achieve desired daylight results simultaneous to building energy use evaluations.
Designer’s Aesthetic Judgment and Form Diversity

An approach to aesthetic evaluation in a generative system was seen in the structural engineering design process of Machwe et al. (2005), in which the authors include aesthetic assessment of design options for a bridge design done through the user interaction with the system. The study suggests integrating user evaluation and aesthetic preference, which has not been widely applied yet in architectural optimization. The work of Brown et al. (2015) introduces a MOO framework that includes aesthetic preference, focused on the structural aspect of building design. They use genetic diversity measure as a fitness function.

Other research on interactive optimization abounds. According to Felkner et al. (2013), some studies have incorporated the aesthetic criterion by using an aesthetic measure of uniformity and the golden ratio, applied to truss structures (Shea and Cagan 1999), or using architectural constraints like the positions of columns in optimizing a continuous roof structure (Pugnale and Sassone 2007). In these approaches, the common difficulty is that aesthetics are subjective to the architect and cannot be straightforwardly quantified (Darke 1979; Felkner et al. 2013). Nevertheless, non-quantifiable objectives have been successfully achieved in interactive evolutionary optimization applied to engineering design, based on the evaluative characteristic of population-based optimization methods (e.g., Takagi 2001; Kim and Cho 2000). Hu and Eberhart (2008) have successfully used the combination of computational power and user-intuitive knowledge, and asserted that the combination is beneficial to achieve complex tasks.

METHODS

In the prototyped design process, the MOGA was applied on energy and daylight performance objectives first. Further, a shape/form comparison algorithm was developed to analyze and rate the final solutions according to a form diversity measure, which potentially facilitates the designer’s aesthetic judgment being used for final design decision.

We divide the experimental protocol into phases. In the first phase, the form-making process is developed, in which the initial building form is parametrically generated as a mass model utilizing the geometric and algorithmic parameters and constraints that allow for form variation. Second, form finding is conducted using environmental analysis, in which the mass model is subjected to performance evaluation, and a search mechanism process is pursued using the MOGA method to optimize the objectives. A shape comparison algorithm is then developed and implemented to eliminate similarities. In the fourth and final phase, a designer evaluation is needed for a decision on whether the final objectives have been met, and the form aesthetics have been satisfied. The design workflow is illustrated in Figure 2.

The system utilizes Rhinoceros, which enables the designer to use different tools. Python, C#, and Grasshopper visual scripting editor were used for modeling and shape comparisons. The environmental analysis uses Ladybug and Honeybee simulation plugins that are based on the environmental analysis tools EnergyPlus, DAYSIM, and OpenStudio (Roudsari et al. 2013).
Octopus was utilized as the MOGA tool (Vierlinger and Bollinger 2014). The following design process was used for developing the prototype.

- The designer sets the input parameters for the form based on the following building design requirements: building type and size, location and weather data, and design objectives. Also, it is necessary to identify the number of spaces and their adjacency relations, with the space adjacency simplified here to just connecting the spaces. To initialize the mass model in Grasshopper, the constraints and variables have to be determined to govern form generation. The form pattern used for the current version is a simple 3D grid based on rectilinear configuration that allows for highly varied design options.

- Once the algorithmic form definition is generated, the mass model is connected to the performance analysis definition using Honeybee and Ladybug tools. Here, windows are added to the mass, and materials are assigned to all building components. Preceding simulation, preparation tasks are needed to connect all required components to Honeybee simulation nodes. Then, fitness functions are assigned for optimal performance: minimal energy use and maximized preferred daylight measures. The analyses are conducted for every design option generated at each optimization iteration (generation). Using the optimization tool Octopus, the designer connects the initial mass, the genepools (parameters), and the fitness function values. Once the optimization run starts, a random population of solutions generates in the solution space, and the run of generations leads to fitter solutions which will eventually form the Pareto front.

- Next, a shape comparison algorithm is developed to limit the solutions to highly diverse forms. It is important to differentiate our suggested form diversity from genetic diversity. Collectively, it is required to maintain genetic diversity to avoid premature convergence, an issue in GAs when the fast progress of evolution toward more fitting solutions in early generations causes convergence that leads to similar solutions in later generations (Renner and Ekärt 2003). The common method used for measuring genetic diversity is the Euclidean distance measure. According to Toffolo and Benini (2003), the concept of distance-based diversity measure relies on the fact that an individual that is distant from all the others has more chances, when mating, to produce offspring in regions of the search space not covered by the current population. Such a distance metric can be calculated mathematically using the (normalized) Euclidean distance in the objective function space, and the measure of diversity of an individual is the sum of its distances from all the other individuals (Toffolo and Benini 2003). If \( p = (p_1, p_2, \ldots, p_n) \) and \( q = (q_1, q_2, \ldots, q_n) \) are two options in the \( n \)-space, then the distance \( d(p, q) \) is given by the Pythagorean formula measured as follows (Deza and Deza 2009):

\[
\text{Distance} (q, p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}
\]

Following this method, the Euclidian distance from the centroid of a set of designs to the furthest outlier was utilized by Brown et al. (2015) to quantify diversity for each optimal set as the following formula: \( D = \max |\text{Centroid} - \mathbf{X}_j| \), where \( \mathbf{X} \) corresponds to the design vector of size \( n \) containing \( n \) design variable settings. For form diversity measure, in our work, we do not calculate the Euclidean distances, rather, we create an algorithm to analyze and compare the shape characteristics of design solutions as explained in the case study.

- Post-optimization, the designer examines the Pareto solutions and evaluates their form qualities visually. The inclusion of elite or near optimal (non-Pareto front) solutions that are aesthetically interesting for the user (or discarding uninteresting solutions) can be decided upon, and further optimization runs can be pursued.

**EXPERIMENTAL CASE STUDY**

The first prototypical experiment has been completed and initial results have been gathered. As a case study, a 1,000 m² office building was selected. The rule for the initial model was mainly to create the building form based on an arrangement of modular units of building spaces, a 3D grid composition. These units can be positioned differently, allowing for flexible modification in the solid/void relation, where solid is the existence of a unit and the void is its absence. Changing the parameters, the adjacency relations among those units change, based on repositioning. The form is customizable to allow for different forms to evolve. For the case study, the prototype was developed according to the following sequence:

1) **Generating the Initial Form**

The architectural layout arrangement used in this experiment is a set of grid squares (units) in a modular 2D grid of 11 x 11 cells, with each unit measuring 10 x 10 m. The unit is 10 m high, and may contain multiple functions and floors. Only 10 units are allowed to occupy the 121-cell grid, leading to 121!/(121-10)! (4.5913309e+20) possible options to emerge in general. We developed an algorithm to allocate each square in a unit-by-unit combinatorial approach. The first unit (Unit 1), was located in the center \((x=6, y=6)\), and the other 9 units are subsequently appended (Figure 3).

The rationale of using the grid system is explained by Michalek...
et al. (2002), who use a similar approach to architectural layout optimization. This layout is simple but can model a large array of architectural layouts, and more complex shapes could be added to the model to expand this array. Additionally, the use of such a grid-searching method, by developing an allocation program to locate the grid units following certain rules, has been applied to architectural space-planning programs (Mitchell and Dillon 1972; Hsu and Krawczyk 2003). It is noteworthy that our configuration of units is similar to a particular family of 2D shapes known as polyominoes, which are geometric shapes constructed by connecting a number of squares edge-to-edge, developed as mathematical games (Golomb 1954; Golomb 1996; Lo et al. 2009). Being three dimensional, our resulting patterns are different from the typical polyomino shapes, particularly when the 3D grid is considered and multiple stories are added to the current layout.

The defined topological constraint here is adjacency-based; each appended unit has to be adjacent to one or more of the existing ones in a combinatorial method. The geometric constraint prohibits overlap or intersection of units at the same location. For Unit 2 for instance, the possible location points are (6-1, 6), (6+1, 6), (6, 6-1), and (6, 6+1) (Figure 3). The process of appending the other units required another constraint, which is to keep the units within the grid boundary. The parameters used are the possible locations of each appended unit that can be changed by the designer using the genepool (genetic slider). Since the location of Unit 1 is fixed, 9 other units’ location parameters are used for the layout optimization. The genetic slider contains values that represent the possible center point locations (x, y) for the appended unit. Once the parameter changes in value, the associated unit is appended in a new location, and the whole layout changes as a consequence. Since the number of possible location points for a new unit can change, a C# program was used to create dynamic genepools that change in their value ranges according to the updated possible locations.

2) Building Performance Evaluation and Optimization
The experiment uses a hypothetical project site, and the weather data of a hot-humid zone was utilized. The parametric mass definition was then prepared for simulation in several tasks.

For the energy analysis, it requires information such as the analysis period, weather file, Honeybee zones, and context. As a program, the building was simplified to comprise mainly two different spaces: the office area and the other spaces (core, lobby, conference rooms, and other services). The window-wall ratio was set to 60% for north and south walls and 0% for the east and west walls. The simulation time-step chosen was for the month of June. For the fitness function, energy use for June was considered, in which cooling loads are dominant according to the climatic information.

For daylight analysis purposes, defining building materials, generating a grid of daylight sensors, and assigning sky conditions were required. From Honeybee zones, building components were retrieved and connected to the simulation nodes. LEED v4 illumination compliance needed to be satisfied, achieving illumination levels (300–3000 lux) for either a ratio of 75% of floor area (2 points) or 90% (3 points) (USGBC 2014). To reduce extended simulation time, the measure was simplified to simulate one hour (12–1 pm) on June 21st for a grid of sensors 5 meters apart.

To connect an objective function to Octopus (the MOGA tool), the fitness value and objective name are required. The energy use objective was set to be minimized, thus the energy use value and name were directly connected to the Octopus objectives. Daylight illumination values were retrieved from the Honeybee analysis node, and the ratio of values that achieve 300–3000 lux over the total area was calculated, and was multiplied by –1 because Octopus always minimizes the objective functions. The illuminance fitness function was calculated as follows:

\[
\text{Illuminance Ratio} = -\left(\frac{\text{Room area of illuminance with 300–3000 lux}}{\text{Total room area}}\right)
\]

To run optimization, the 9 genepools were connected to the genotypes, the mesh of the initial mass model was linked to the phenotypes, and the fitness functions were linked to the objectives in Octopus. Next, a random design population was
initiated, and for every design option in the solution space, daylight and energy analyses were conducted. The MOGA run searches for fitter solutions; when daylight and energy performance is improved, forms of design options evolve and change in composition. The Pareto front, Elite, and history solutions of the optimization run at generation 10 are shown in Figure 4 and explained in the discussion section.

3) Shape Comparison Algorithm
An example shape comparison algorithm for this case study was created to compare the solutionss when the optimization run was completed. The algorithm first records all meshes and unit center points for 100 of the Pareto front and Elite solutions as illustrated in Figure 1. The Elite set refers to the best fit solutions that are guaranteed as parent candidates in the next generations; when a new solution set emerges, it is compared to the worst of the elite set, and only if the new solution is better will it replace the worst Elite solution (Musnjak and Golub 2004). Next, the solutions were subjected to a form diversity measure, calculated by identifying the overlap of unit center points among the solutions (after alignment). We coded a Python program to do two main operations. When comparing two solutions, first, 10 different translations were used to move and align the first unit’s center point of solution 1 to each unit’s center point of solution 2. Second, a comparison of the unit’s center points between the two solutions is made for each of the 10 translations, and the smallest number of non-overlapping units, the minimum (shape difference), is recorded as the actual difference between the two solutions. For example, Figure 5 illustrates how the first 10 shape difference values were computed for aligning Unit 1’s center point of Solution 1 (red) to each of the 10 units’ center points in Solution 2 (gray), respectively. The value that represents the minimum difference value and indicates the highest overlap of the two solutions was stored as the shape difference value of 4. Each solution was paired and compared with all other solutions; the comparison was applied to the 100 solutions in a cross-reference operation (100x99/2). The minimum shape difference value for each pair was used as a measure for form diversity. As recorded, the shape difference values were between 1 and 8 in the optimization experiment run. Solutions of higher shape difference values were retained (more values above 5), and solutions of low difference values (more values below 5) were eliminated. Out of 100 solutions, 20 diverse forms resulted after shape comparison (Figure 6).

4) Designer Interaction
Forms retrieved after shape comparison facilitate assessment and designers’ input. Ending the simulation run is based on the designer’s decision after examining the numerical thresholds and form qualities.

DISCUSSION OF EXPERIMENTAL OUTCOME
For demonstration and discussion purposes, the solution space of generation 10, of which Pareto and Elite solutions (100 solutions) were selected to be recorded and stored in Grasshopper from an evaluation pool of 191 solutions (Figure 4). Octopus is based on the Strength Pareto Evolutionary Algorithm (SPEA-2), an improved elitist multi-objective evolutionary algorithm (Vierlinger and Bollinger 2014).

The solution space (Figure 4) shows the Pareto-front solutions as a curve, and the two objective function values in the 2D solution
Ten possible overlap options for a pair of two solutions with associated shape difference values, and the minimum shape difference value highlighted.

Above: all 100 selected solutions for shape comparison. Below: 20 solutions retrieved after shape comparison.
space. It can be seen from Figure 4 that the solutions started to define a Pareto front in generation 10. The solutions are satisfactory and worthy of discussion. As marked, two Pareto solutions were selected and visualized for discussion.

Solution #1 (Figure 7) is constituted by a pattern (configuration) approximately aligned along the north-south axis. For the east-west facades, no windows exist, thus in such a pattern, minimum glazed surfaces are exposed to thermal conditions. This explains how Solution #1 satisfies the minimal energy use (21357 kW/h) by avoiding windows. As for daylight illuminance, Solution #1 is one of the worst options, with an illuminance ratio of 0.375. Such a result illustrates how conflicting the two objectives of energy and daylight are.

For Solution #2 (Figure 8), the illuminance objective has been successfully achieved by a pattern approximately aligned along the east-west axis, with lots of windows, allowing maximum daylight exposure (the illuminance ratio is 1.00). Nevertheless, this solution is worse in terms of energy use (24073 kW/h). The Pareto-front results illustrate an expected response to satisfy the two conflicting criteria, and this verifies that optimization was working successfully.

As for the shape comparison, the algorithm was able to identify the most diverse solutions in terms of building forms among all the 100 best solutions from the optimization run.

CONCLUSIONS & FUTURE WORK

Though simple, the case study shows the potential for the system to enact the designer’s control of the optimization process, particularly for diverse forms’ evolution and assessment. The case study in this paper uses adjacent boxes to simplify the design problem for the purpose of demonstrating the main idea of incorporating form diversity in design optimization. Suggested improvements include defining adjacency constraints according to the building’s programmatic requirements. Importantly, other architectural requirements need to be addressed and incorporated into optimization, particularly the requirements related to energy modeling and daylight evaluation such as zoning, cores, and mechanical, electrical, and plumbing (MEP) design requirements.

For shape comparison, it can be further developed and applied to more complex three-dimensional forms for multiple stories and high-rise buildings (Figure 9). Potential improvements in shape comparison include four aspects. First, although the utilized translation procedure was successful to guarantee that similar solutions that differ only in location are eliminated, there is a need to add rotation and mirror procedures to delete other similar shapes. Second, clustering of similar shapes in groups, e.g., using the k-means algorithm, will be important to incorporate in the shape comparison process. Third, shape comparison can be integrated earlier, before conducting environmental evaluation, to save computation time. As a fourth aspect, it is important to test developing a fitness function for form diversity as an objective to be satisfied in addition to the daylight and energy objectives. As such, the generative system will be complex as the shape analysis is not internal to a single solution but is an analysis among all design solutions.
As for the designer’s aesthetic assessment, a score-rating algorithm can be developed and integrated to rank the solutions according to the designer’s preference, which is a concept that will be implemented in the next version of the prototype. Validation of this prototypical version was a simple comparison between the 100 solutions obtained before shape comparison and the 20 solutions retrieved after shape comparison. Testing the results, the algorithm has led to a much smaller number and still significantly diverse building forms. In the future, prototypes of more complex design case studies will be pursued and validated.

ACKNOWLEDGEMENTS
The authors thank the BIM-SIM Group, Chengde Wu, Prof. Philip Galanter, Dr. Stephen Caffey, and Department of Architecture, TAMU.

REFERENCES


IMAGE CREDITS
All images by Shermeen Yousif.

Shermeen Yousif is a Ph.D. candidate, and Professor of Record in the Department of Architecture, Texas A&M University. She received her Master’s Degree in Architecture from Dessau Institute of Architecture, Anhalt University, Germany where she was awarded the German Academic Exchange Program DAAD scholarship. She has also received multiple grants and taught several courses and design studios in architecture. Her research focus is on generative systems, Multi-Objective Optimization, building form aesthetics, parametric modeling, and building energy simulation and daylight analysis.

Dr. Wei Yan, Professor, Presidential Impact Fellow, Texas A&M University, studied at Tianjin University, ETH Zurich, and University of California, Berkeley (Ph.D. in Architecture and M.S. in Computer Science). His research projects in Design Computation and Building Science are funded by NSF, NEH, DOE, NSF China, ASHRAE, Autodesk, etc. He has served as a reviewer for NSF, Hong Kong Research Grant Council, and Austrian Science Fund. He received a Best Paper Prize in 2006 International Conference on Design Computing and Cognition. In 2012, he was a Guest Scientist at Lawrence Berkeley National Laboratory and a Visiting Professor at Stanford University.

Dr. Charles Culp is a Professor, Texas A&M University. He focuses on energy efficiency and comfort in buildings with over 40 years of industry and academic experience. He is currently focused on developing methods to improve the comfort, health and energy efficiency of buildings and is working on prototyping a high efficiency non-Carnot cycle based air conditioning technology that he co-invented.