Abstract. This work introduces a new system in architectural design optimization that integrates form diversity and clustering methods into the process. The first method we propose is an algorithm for rating design solutions according to their geometric correspondences, maximizing differences and enforcing diversity. In addition, we implement the K-means algorithm to cluster the resulting design forms into groups of similar forms, to substitute each group with one representative solution. The work aims to facilitate decision making and form evaluation for designers, leading to an interactive optimization process, and contributing to improving existing optimization models in architectural design research and practice. We modeled a dynamic system through prototyping, experimenting and test-case application. As a prototype development, the protocol was done through phases of: (1) parametric modeling, (2) conducting energy simulation and daylight analysis and running a generative system, and (3) developing an algorithm for form diversity and another for implementing K-means clustering. The results are illustrated and discussed in detail in the paper.

Keywords. Architectural Design Optimization; Form Diversity; K-Means Clustering.

1. Introduction

Architectural design can be treated as an optimization process in which finding optimal solutions that satisfy predetermined objectives is targeted (Yan et al. 2015). A continuous thread of research has been conducted on architectural design optimization and generative systems to evolve performance-driven building design solutions. Despite the potential in advancing the design process of efficient buildings, optimization models still suffer multiple issues (Attia et al. 2013). One issue in available optimization platforms is that appraisal of form qualities is not explicitly incorporated in the process. Design optimization has to integrate building form evaluation in addition to other performance-based criteria to guarantee that a full definition of a building is developed (Yi et al. 2015). Another issue is the too many, and sometimes redundant design solutions often generated from the search mechanism (e.g. hundreds), and they can be similar in form due to the possibility of slight changes in model’s parameters. The overwhelming number of solutions causes designers’ fatigue,
inhibits effective decision making, and leads to inefficient interaction between the designers and the optimization system. This work introduces an innovative system in architectural design optimization that integrates form diversity, facilitating architects’ evaluation of design solution forms. The first objective is to develop an algorithm for form diversity to rate design solutions according to their geometric differences, and to exclude similar forms. Another solution to the excessive number of solutions is to cluster the design solutions resulting from optimization into groups. This enables designers to perceive the grouped forms, and to have a control on manipulating the number of clusters (to decrease the number) of final design solutions. The primary contribution of this work is a simple, yet robust algorithm for finding correspondences between geometric forms, maximizing difference sand enforcing diversity. The aim is to apply this algorithm to other forms, more complex test-cases, and test the potential generalization of the method. Ultimately, the contribution is to computational design methods through accommodating designers’ aesthetics and form appraisal into architectural design optimization.

2. Background

Recently, progress towards developing optimization methods has been investigated and integrated into architectural design research (Aish et al. 2012). Several studies have been published in the field that proposed or experimented with the use of optimization to create building form alternatives (Caldas and Norford 2003, Gerber et al. 2012, Welle et al. 2012, Bradner and Davis 2013, Konis et al. 2016). For Building (2D) shape optimization, Wang et al. (2006) have employed a Multi-Objective Genetic Algorithm (MOGA) to optimize building layout and orientation. A commonality shared by these studies is the focus on producing variations of building forms, informed mainly by energy footprint or/and other quantifiable criteria, without considering issues of form diversity or the pursuit of eliminating similar forms. The system we suggest here includes tasks in the order of (1) parametric modeling, (2) energy simulation and daylight analysis, to be run during optimization, (3) in addition to post-optimization articulation of solutions utilizing form diversity and K-means clustering algorithms. The following is an explanation of technical areas and methods that are utilized in this work.

2.1. PARAMETRIC MODELING

Parametric Modeling is a method to define models with constraints and variable parameters utilizing rules to automatically modify design options based on changing parameters (Aish and Woodbury 2005). One of the most intelligent uses of parametric modeling is related to exploring new forms, and allowing for a wide range of building efficiencies, such as structure, energy, construction, and circulation (Caplan 2011).
2.2. ENERGY SIMULATION AND DAYLIGHT STUDY

In a performance-based design approach, energy simulation typically needs to be coupled with daylight studies for an accurate whole building simulation (Roudsari et al. 2013). Simulating every design solution, a maximized preferred daylight measure and minimum energy use are targeted in this study as the quantifiable performance objectives that need to be satisfied.

2.3. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM

To solve Multi-Objective Optimization (MOO) problems, two methods are typically followed. The first is the weighted sum method which is to utilize a composite function of the sum of all objectives, each given a weight. The second method is the Pareto Front, which is to find the set of non-inferior optimal solutions (Radford and Gero 1987, Konak et al. 2006). A Pareto optimal solution is retrieved through an optimal trade-off between two or more objectives, where an objective cannot be improved without the other being degraded (Vierlinger 2013). A Multi-Objective Evolutionary Algorithm (MOEA) approach is a method to solve optimization of multiple objectives utilizing the principles of evolution, through search space exploration in a generative system to find the fittest solutions (Vierlinger 2013). This work utilizes a MEOA-based tool for Pareto optimization.

2.4. FORM DIVERSITY

Often, genetic diversity is used to penalize closely-distanced solutions in the solution space to escape the local optima. Genetic diversity is a distance-based measure that can be often calculated through measuring the Euclidean distances among all individuals in a population in the solution space; the diversity metric of one individual is the aggregation of its distances from all of the others (Toffolo and Benini 2003). In the work of (Brown et al. 2015), a structural design optimization model, the search for diversity in solutions and seeking interaction with the performance feedback has been made utilizing genetic diversity. Unlike those works that focus on the genetic diversity of solutions’ parameters in the genotypical space during optimization (which can be dimensions, materials, room connectivity, etc.), our work is focused on diversity of forms by comparing the geometric correspondences among the candidates post to optimization, considering only the form parameters. For instance, when a rotation of a design solution is a parameter, multiple rotated solutions mean multiple different solutions (different in parameters) in terms of genetic diversity, while they all mean one form in terms of our form diversity. In a prior work, a form diversity algorithm has been developed through creating a program to rate resulting forms according to their geometric differences, as the algorithm discards similar forms, and retains highly diverse candidates (Yousif et al. 2017).

2.5. K-MEANS CLUSTERING

The concept of cluster analysis becomes important for understanding meaningful groups that share mutual characteristics; collectively, cluster analysis is important for variety of areas, such as pattern recognition (Tan 2006, Tan et al. 2013).
K-means clustering is an algorithm to divide a matrix of observations or data (like points) in N dimensions into K clusters in which the within-cluster sum of squared errors is minimized (Hartigan and Wong 1979). The main steps of K-means algorithm can be explained as follows: (1) selecting the initial division with K clusters, and repeat steps 2 and 3 until cluster membership stabilizes, (2) creating a new partition by assigning each point to its closest cluster center, and (3) computing new cluster centers (Jain and Dubes 1988, Jain 2010). The rationale behind using the K-means clustering method in our work is because it is a promising approach to group the design solutions into sets of similar forms, then to find the representative solution candidate of each cluster to be presented to designers, eliminating similarity, and reducing the number of final solutions. In concept, in our research, for a set of design solutions in the solution space of a generative system, the search is for clusters (the set of solutions that belong to each cluster), analyzed according to their parameters of geometric forms. In addition, it would be beneficial to find the centroid of each cluster, then the representative candidate as the closest solution to that centroid (Kanungo et al. 2002). Although K-means utilizes the Euclidean distance measure, we apply it only on geometric parameters after optimization, as explained in the following section.

3. Methods
To illustrate the system we suggest, we created a dynamic model of the system workflow, through prototyping, experimenting, and test-case application. In prototyping, a series of procedures and steps were conducted and tested to develop a functional prototype for optimization, and for developing and implementing the form diversity and K-means clustering methods.

3.1. PROTOTYPE DEVELOPMENT
We followed the protocol illustrated below in Figure 1 for prototype development. In phase one, we set up the initial model using parametric modeling, in Grasshopper/Rhino as the main platform. Next, we prepare for energy simulation and daylight analysis utilizing Honeybee and Ladybug plugins, and conduct optimization of those objectives using the Multi-Objective Evolutionary Algorithm tool (Octopus) as the search mechanism. In the third phase, post to optimization, we created and implemented a form comparison algorithm using Python and programmed an algorithmic definition to apply K-means clustering analysis on resulting forms, using the Owl plugin nodes. The prototype has been developed considering early design phase, although it can be applied to other phases. Multiple tools were used for the prototype, as indicated in Figure 1.

3.2. TEST-CASE EXPERIMENT
To illustrate the prototype, we have utilized a test-case in which we conducted the following processes:
3.2.1. Creating a Grid-Based Pattern

In a previous work, a simple grid-based pattern was created that relies on variable allocation of units (boxes) to a cellular grid, allowing for 4.5913309e+20 options to be generated, creating patterns such as in Figure 2. The rationale for using such a heavily restricted grid-based layout is due to its simplicity yet capability to generate a wide range of design options (Michalek et al. 2002), that are required for testing the form diversity algorithm (Yousif et al. 2017). Another reason for utilizing this layout is its capability to produce typological design options. Typological designs refer to designs that represent typical typologies of building configurations such as L-shapes and U-shapes, considering the fact that designers do not utilize pure typologies, rather mixtures of typologies (Wurzer et al. 2017). Our layout was set up using a grid of (11x11) cell, in which 10 units (boxes) are allowed to occupy the grid. The first unit was allocated in the center of the grid (x=6, y=6), and 9 units were appended consequently, allowing their center points to changes as parameters. By coding customized definitions, three algorithmic constraints were used in this layout: (1) adjacency, each unit must be adjacent to at least one side of the previous unit/s, (2) non-overlap, a unit cannot be located on an occupied cell, and (3) boundary-respect, the grid boundary has to be satisfied, and units cannot be appended outside the grid. Samples of the resulted forms are illustrated in Figure 2. The grid-based setup was meant to be a simplified design form to test and experiment with the form diversity and K-means cluster analysis, acknowledging that building designs are often more complex.

3.2.2. Setup Performance Evaluation and Optimization Run

To conduct performance simulation while the MOEA runs, a preliminary task is needed, in which the setup of simulation requirements is made. To prepare for energy simulation, the geometry of the initial model was converted into zones, and a program of an office space was attributed to the zones. Next, glazing ratios were selected to each façade, with 60% Window-To-Wall (WWR) ratio for the north and south facades and 0% WWR for the east and west façades. Weather file of College Station, TX (TMY3 format), simulation time-step, and schedules were added to the energy simulation definition. Cooling loads were selected to be the main loads to minimize for the energy fitness function because it is typically challenging to minimize the cooling loads while maximizing daylight illuminance for the hot humid climate selected here as the two criteria conflicts. For the daylight study, materials were added to prescribe the building components: walls, floors, and ceilings. The spatial Daylight Autonomy (sDA), which is a measure of illuminance levels that needs to be achieved for LEED v4, was selected as a metric; it has to be between(300-3000) lux for 9 a.m. and 3 p.m., both on a clear-sky day at the equinox (USGB 2014). A grid of daylight sensors was created according to the floor plate of the initial model; the sensors were distributed 5 meters apart, to save computation time. A standard sky generator in addition to an exterior surface, adjacent to the building model were connected to the grid-based simulation. The maximum ratio of the area that achieves 300-3000 lux over the total area was pursued as the fitness value.
To run the simulation-based MOEA algorithm, the mass model, the 9 parameters, and the fitness values (minimum cooling loads, and maximum sDA ratio) were connected to the optimization tool (Octopus). For every generation, energy simulation and daylight simulation are conducted for all design candidates (evaluation pool size was 191). In some solutions, the form pattern took north-south orientation to maximize daylight illuminance (e.g. solution 2) which the solution has successfully achieved, while it is considered one of the solutions of high cooling loads. In other cases, forms are east-west oriented with minimum windows to minimize cooling loads (e.g. solution 1) with low illuminance values, as shown in Figure 3. The Octopus solution space, illustrated in Figure 4, shows the results of optimization at generation 20. The Pareto front is indicated as a curve with 8 solutions, represented as meshes, and the Elite solutions (the best solutions to potentially survive in the next generations) are also shown as meshes, while solutions of previous generations are represented as dots (history). All of the Pareto front, and 12 of the Elite solutions were marked (in bubbles) and reinstated into Grasshopper for K-means clustering. The optimization results of this experiment confirm an expected behavior to satisfy the two conflicting objectives, and prove the algorithmic run successful, with the results of solution 1 and 2 as examples.

3.2.3. Develop and Implement Form Diversity Algorithm and K-means Clustering

3.2.3.1. Form Diversity Algorithm

In a prior work, post to optimization, a hundred of the Pareto front and the Elite solutions were retrieved from generation 10 of a different optimization experiment and were subjected to form comparison. From 100 solutions, the algorithm filtered 20 of high difference values, thus of high form diversity (Yousif et al. 2017). To compute form diversity as a measure, the form comparison algorithm was created to perform four procedures as follows. In (1) pairwise-comparison & translation (move), every design solution was paired with all of the other 99 solutions, and for each pairwise comparison, 10 translation operations were made to find the maximum overlap and the minimum difference value. Step (2) is to record the form difference values; the minimum difference value was considered as the actual difference value for each pair compared, and thus was recorded. Next, in (3) cross-reference matrix, the difference values between a solution and the other 99 were stored for each solution in a cross-reference matrix. Finally, in step (4) filtering solutions of high difference values occurred, where solutions with higher difference values were retained and solutions with lower difference values were discarded (Yousif et al. 2017).

3.2.3.2. K-means Clustering Algorithm

At generation 20, the Pareto front (8 solutions) and 12 selected Elite solutions, retrieved from optimization, were subjected to multiple procedures to prepare the required parameters to the K-means algorithmic definition. The K-means algorithm from the Owl tool requires tensors as the primary parameter, in addition to the number of clusters, number of K-means iterations (was set to 100), and
seeds (was set to 2399). Tensors refer to mathematical objects, used to describe physical properties similar to scalars and vectors; a scalar is a zero rank tensor, and a vector is a first rank tensor (Kline 1990). In the Owl tool, according to the author, the K-means algorithm was created to accommodate multiple dimensions or coordinates, N (numbers) of dimensions, and in our test-case we have 10 dimensions representing the 10 center points of each design solution. The next required task was to convert the ten center points of the ten units of each design solution into tensors in 10-dimensional K-means clustering. As a result, the K-means definition provided the clusters, and indices of the list items that belong to each cluster. It is important to note that the K-means algorithm was applied here on the 20 solutions overlapping, as in the left side of Figure 5. First, as a sample, the number of clusters was set to 2 (case 1), and the result was two clusters of 11 and 9 solutions, as shaded differently (black and grey) in the upper part of Figure 5. Next, we changed the number of clusters to 3 (case 2) and retrieved three different groups (shown in black, grey, and light grey), that can be seen in the lower part of Figure 5. To show their forms, the solutions of both cases were re-oriented, and distributed along an array of 20 cells, as in the right side of Figure 5.

Figure 1. Workflow of the prototype, representing a combination of several processes. The major addition to current optimization models is process 3.

Figure 2. Axonometric view of 8 possible options of the grid-based pattern.

Figure 3. From left to right: 3D view, top view, and illuminance mesh of solution 1 and 2.
4. Discussion of Test-Case Outcome

The results of K-means clustering are preliminary, and more experiments are needed to verify the findings. Often, the results of K-means clustering can be best represented in a solutions space, which is typically a two or three-dimensional space, but here it cannot be directly illustrated as the dimensions were ten. The solutions are analyzed and compared when they overlap (with the grid-center unit overlapping in all solutions), and this overlap is thought to be reliable for accurate comparison of the grid-based patterns. It is obvious that more center points overlapping means similar forms, and leads to cluster their forms together. There was no mirror, move, or rotation procedures here to recognize more similar patterns. Comparing the two cases pursued (case 1 and case 2), it can be noted that the clusters were formed slightly differently. This behavior can be attributed to the inherent characteristic of K-means that alternates between updating the assignments of solutions to clusters (Raykov et al. 2016). The outcome of the Owl-based K-means definition used here was mainly the clusters and their assigned members. Another expected outcome was the centroid of each cluster, which has not resulted. Therefore, the representative candidate of each cluster could not be retrieved in this experiment. Adding the task of obtaining the centroids to the current algorithm is important as a next step for this work. It is
important to note that in a typical K-means clustering, the greater homogeneity within a group, the greater the difference between groups is, leading to more distinct clustering (Tan 2006, Tan et al. 2013). This characteristic of K-means is important and advantageous to the form comparison targeted in this study.

5. Conclusions and Future Work

This work is part of an ongoing research that investigates incorporating form diversity and K-means cluster analysis to create a new architectural design optimization system. Although a simple test-case was used, the form diversity algorithm has successfully led to filtering highly diverse forms. The initial results of K-means clustering are consistent with the characteristics of the typical K-means method, yet, verification procedures need to be conducted to compare the results across different scenarios and test-cases. This work aims to contribute to extend the capabilities of architectural design optimization by creating a method to be applied to general cases of building forms. Eventually, the method will be an open-source program, accessible and connected to available modeling and optimization platforms. In parallel to this work, in a recent publication, we have investigated the possibility of incorporating architectural aesthetic variables of modern language as constraints and parameters into initial modeling to enhance the aesthetic quality of design solutions generated from optimization (Yousif et al. 2018). The aesthetic variables will be incorporated into our system, in addition to form diversity and K-means algorithms.

References


Tan, P.N.: 2006, Introduction to data mining, Pearson Education India.


