Advances in Computational Design, *Vol. 2, No. 3 (2017) 241-256*
DOI: https://doi.org/10.12989/acd.2017.2.3.241

Genetic algorithms for balancing multiple variables in design practice

Bomin Kim[1a] and Youngjin Lee[*1,2]

[1]Architecture, Sasaki Associates, 64 Pleasant Street, Watertown, Massachusetts, 02472 United States
[2]Department of Architecture, Boston Architectural College, 320 Newbury Street, Boston, Massachusetts, 02115, United States

(Received July 13, 2017, Revised July 20, 2017, Accepted July 21, 2017)

Abstract. This paper introduces the process for Multi-objective Optimization Framework (MOF) which mediates multiple conflicting design targets. Even though the extensive researches have shown the benefits of optimization in engineering and design disciplines, most optimizations have been limited to the performance-related targets or the single-objective optimization which seek optimum solution within one design parameter. In design practice, however, designers should consider the multiple parameters whose resultant purposes are conflicting.

The MOF is a BIM-integrated and simulation-based parametric workflow capable of optimizing the configuration of building components by using performance and non-performance driven measure to satisfy requirements including build programs, climate-based daylighting, occupant's experience, construction cost and etc. The MOF will generate, evaluate all different possible configurations within the predefined each parameter, present the most optimized set of solution, and then feed BIM environment to minimize data loss across software platform. This paper illustrates how Multi-objective optimization methodology can be utilized in design practice by integrating advanced simulation, optimization algorithm and BIM.

Keywords: genetic algorithm; multi-objective optimization; parametric and evolutionary design; BIM

1. Introduction

High-performance architecture has been using optimization to find the building form, orientation, window size and etc. relating to a design target to reach optimized performance. There have been, however, few researches that optimize the conflicting multiple variables, in particular the performance-based measure and non-performance driven design targets because they provide a direct conflict in the design process, which make it more difficult to satisfy multiple constraints in architectural practice. But the non-performance driven design targets such as occupant's view experience and building cost are critical aspects in practical design decision, and in some cases they are considered more important than performance-based measure. When considering multiple variables, compromises must be made in the context of contradictory interests. The complex

∗Corresponding author, Profesor, E-mail: youngjin.lee@the-bac.edu or ylee@sasaki.com
[a]Associate AIA, CPHC, E-mail: bkim@sasaki.com

http://www.techno-press.org/?journal=acd&subpage=7 ISSN: 2283-8477 (Print), 2466-0523 (Online)

networks of dependencies and correlations that often exists between these interests often restrains the designer and their ability to find an optimal solution. Such conflicting variables mentioned above such as daylight, construction costs, occupant views, etc., can be major opponents in the design of a building which need to be analyzed holistically, especially in their relations to each other.

In recent years, architects' techniques and design tools used to approach design practice have changed significantly. With the adoption of contemporary technology, the regular use of BIM as a process for creating and managing a project's information has given way to analytical approaches of quantification and computer aided design technologies. Such adaptations allow for a more flexible process of design and storing building information, as well as creating platforms for numerical methods which enable quicker analysis of more complex systems. (Hoffman 2013) We've seen wide adoption of integrated parametrically defined analysis modeling techniques allowing for parametric associative modeling techniques where representation is generated by a set of rules that relate to one another (Thiele 2009). Such data includes everything from geometric models, BIM, to structural, solar modeling-anything that can be analyzed and represented within a clearly defined metric. These methodologies dependent on the very data that is easily altered as inputs create a desire for a single process generation utilizing a set of compatible variables, a process that is still cumbersome and not utilized. As an application based on Autodesk's Revit, the most widely used BIM software in practice, generation of complex geometry is limited and data representation is prescribed to application settings. There are also limitations for modeling large-scale projects such as master planning design (Roudsari *et al*. 2013). In addition, such environmental analysis software packages, have limitations in certain conditions including the loss of BIM data. On the other hand, Rhinoceros, a popular CAD program's lack of BIM modeling capabilities makes optimization exploration significantly faster making it an ideal platform for environmental analysis, but requiring additional steps for documentation.

Given two limitations mentioned above, this paper proposes Multi-objective Optimization Framework (MOF). MOF is a BIM-integrated and simulation-based parametric workflow capable of optimizing the configuration of building components by using performance and non-performance driven measure to satisfy the requirements including build programs, climate-based daylighting, occupant's experience, construction cost and etc. Traditionally, the design process to realize these variables requires a designer's experience and intuition, often relying on trial and error approaches that become inappropriate beyond similar design parameters including program, typologies, and climates. Instead, the MOF will generate, evaluate all different possible configurations within the each predefined parameter, present the most optimized set of solution, and then feed BIM environment to reduce the amount of time and information lost from model sharing while optimizing multiple variables. Compared to the workflow of other findings that intended to assist designers in the early stage of design, MOF can be applied to any stage of design depending on the design parameters designers set. This paper illustrates how Multi-objective optimization methodology can be utilized in design practice by integrating advanced simulation, optimization algorithm and BIM. Section 2 explores the literature of optimization techniques and the imminent demand of multi-objective optimization in architectural practice. Section 3 describes the genetic algorithm as an optimizer for this research. In Section 4, two case studies show the overall workflow of MOF and demonstrate how MOF is applied to the architectural practice. Finally, a discussion on the future research direction is provided along with a conclusion.

2. Optimization and optimizer

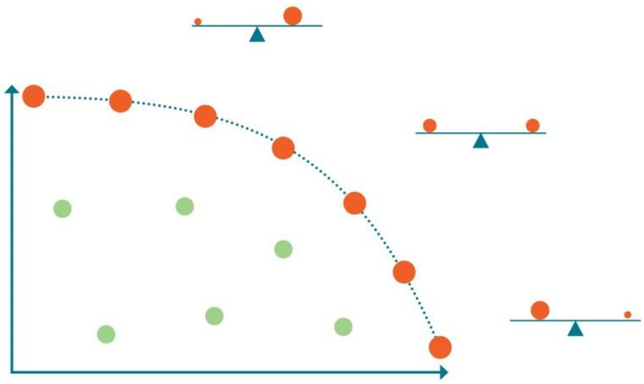


Fig. 1 Pareto-front

2.1 Single-objective vs. multi-objective optimization

Optimization means to seek minima or maxima of a function within a given defined domain (Hofmann 2013). Among several types of techniques for architects to discover an “optimal” solution, “passive” optimization is what many designers think they are doing by creating three of four options, mentally testing them against past experience, and then intuitively determining a “best” solution (Sukreet and Kensek 2014). Although some architects can be very good at this, especially if they have a lot of experience with passive strategies, sometimes their intuition is wrong (Konis *et al*. 2015). To be able to make a justified decision, the designer needs to know about 1. The extreme trade-off configurations and 2. The relations that are causing the necessity for trade-offs. Traditionally, the process to gain this knowledge relies on a designer’s experience and, depending on the complexity and uncertainty of the task, additional trial-and-error to unveil the relational nature of a design’s sub-problems (Thiele *et al*. 2009). In order to achieve the reliable results that are not based on designer’s intuition or experience, a computer-aided simulation method will be safer mode. From that sense most recent researches utilized computer-aided optimization method to generate and evaluate all different possible configurations within the predefined limits of each parameter, and present the most optimized set of solutions to achieve high-performance building. Most of them, however, are the single-objective method which addresses a single variable, e.g., building orientation for minimum energy load, fenestration configuration for maximum daylight and etc. A vast number of factors determine the final design of a building. Many of them in general conflict between different targets. In all affairs, beyond the realm of design alone, trade-offs have to be accepted in the context of contradictory interests. A complex network of dependencies and correlations between interests exists, which restrains, in our case, the designers in finding a good solution easily (Vierlinger and Hofmann 2013). Multi-objective optimization, MOF, can be applied to the case where optimal decisions need to be made in the presence of trade-offs between two more or conflicting objectives respectively. In practical design process where multiple objectives need to be explored, multi-objective optimization is considered more relevant than single-objective optimization. When balancing multi-objective optimizations, Pareto-optimality of a solution occurs when an optimal trade-off between two or more contradicting objectives, where one goal cannot be improved without degrading the others. The theoretical sum of all Pareto-optimal solution is called the Pareto-front (Fig. 1), which can be thought of as a hypersurface in n-dimensional space where n is either the number of parameters of

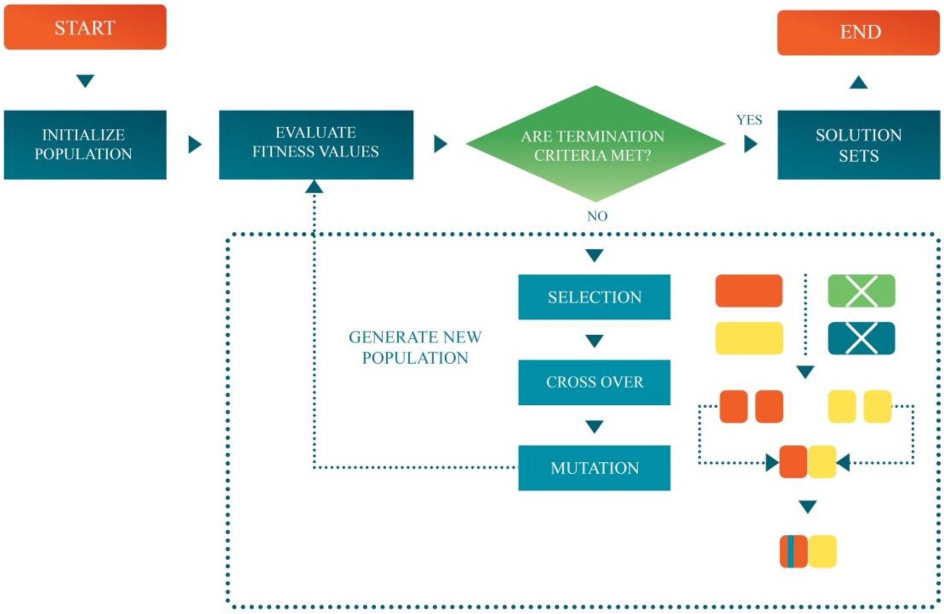


Fig. 2 Overall workflow of Genetic algorithm

the number of objectives respectively (York 2013). Since the parameter and objective space can correlate in highly complex ways, a main effort of algorithms for search and optimization is to overcome non-linear relations imposed by the algorithmic definition, which connects the parameter to objective space (Bader 2010, Zitzler 2001).

2.2 Genetic algorithm as optimizer

The Genetic algorithm (GA) is utilized as an optimizer of multi-objective optimization in this study. GA mimics the theory of evolution employing the same trial-and-error methods that nature uses in order to arrive at optimized results. When automated for specific parameters and results, this technique becomes an effective way to computationally drive controlled results within the iterative design process-allowing designers to produce optimized parameters resulting in a form, graphic or piece data that best meets design criteria (Kilian 2006). Analogous to an open-ended design process, the evolution process of a GA never reaches a final state, but also at each point yields something that can be considered a result (Vierlinger and Hofmann 2013). This section will briefly introduce the concept of GA in order to provide the better understanding on how GA can fit into MOF within the context of architectural practice.

GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GA operates on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution (Chipperfield *et al*. 1995). At each generation, a new set of approximations is created by the process of selecting individuals, i.e. chromosomes, according to their level of fitness in the problem domain, and breeding them together using operators borrowed from natural genetics. This adapted GA process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were derived, just as in natural adaptation. Then the GA process can

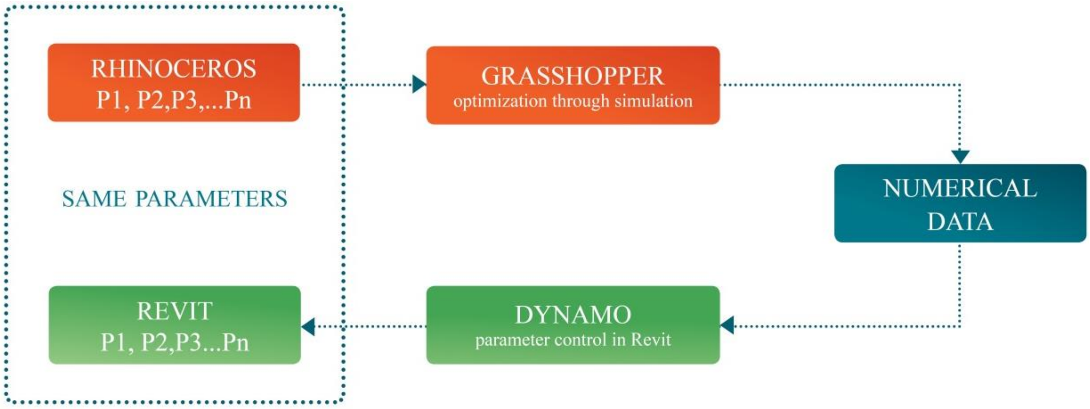


Fig. 3 Overall workflow of multi-objective optimization framework

be subdivided into three parts: population, evaluation, and reproduction. It begins with an initial population that may be a random combination of individuals. Next, it goes through a process in which each individual in the population group will be evaluated. The GA then assigns every individual a grade based on its characteristics-also called the fitness function. This fitness will be taken into account in order to decide each population member's eligibility of reproducing. Generally, the better performing individual with a higher correlating fitness function will adapt to the new generation more than lower performing individuals. Part of the initial population gives place to the new population; this process is called "breeding" of the initial population. The cycle will continuously run until some numbers of generations are completed or until some condition is satisfied as shown in Fig. 2.

Though the evolution process of a GA never reaches a final state, it converges and yields a optimal solution. The evolutionary optimization algorithm runs through different random iterations, looping through and selecting the best-fit individuals for reproduction, breeding through different operators, evaluating the fitness of new offspring and replacing a part of the population with the fittest offspring until the convergence criteria is reached (Vierlinger and Hofmann 2013). Genetic algorithm can be used for complex design, a problem that requires many iterations, producing a myriad of variations and alternatives and provide graphical methods of evaluation. "Using these tools, the parametric system becomes the genomes, the fields of alternatives becomes the population, and the architect's design goal becomes the fitness criteria" (Miller 2011).

3. Overall workflow of MOF (multi-objective optimization framework)

Design methods in the built environment are subject to constant change and evolution requiring techniques and tools developed to be highly adaptable and applicable to other parametric modeling platforms. MOF looks to bridge these needs by

1. Developing multi-objective strategies in parametric design
2. Create an adaptable interface depending on design case
3. Implementing and evaluation of performa

As described above, Revit, the most widely used BIM software in practice excels in documentation and tracking the necessary information, but it lacks the capability of generating

Fig. 4 Example of kinetic shading device

Fig. 5 Example of fixed shading device

complex geometry as well as evaluation through analyzing real-time simulations in parametric model. Rhinoceros along with its visual scripting platform Grasshopper and other native and non-native plug-ins, is superior in analysis through simulation in parametric model and generation of solutions, however, lacks in BIM and documentation, requiring significant efforts for adequate project documentation.

MOF intends to integrate the benefits of the two different platforms as shown in Fig. 3. Firstly, the same geometry is shared between Revit and Rhinoceros by exporting geometry from Revit to Rhino in a mesh format. Next, the parameters to control the geometry of architectural components are set up in Revit and Rhinoceros as appropriate. The following step is to create a compatible parametric model with those parameters by using Grasshopper to take advantage of the platform's superior parametric environment which enables different modifications without the need of regenerating the entire model every time. This study utilizes Octopus, a plug-in of Grasshopper as a multi-objective optimizer as well as other plug-ins for environmental analysis tools such as Ladybug, Honeybee and DIVA depending on the design specific targets. The optimal solution through simulation based on fitness function is presented in numerical data that can be extracted to Excel. The numerical data in Excel are then fed into Revit via Dynamo, a visual script language for Revit to control the parameters of Revit Family components. The numerical data result in the same behavior in Revit Family component as those in Grasshopper because both models utilize the same parameters set up in Revit and Rhino during the first and second step. This prevents data loss during modeling, simulation and analysis across the two different platforms. In the following section, we will explore how MOF works to illustrate the principles during specific practical examples.

4. Case studies

4.1 Case study-1: Interior illumination vs. occupant's view experience

4.1.1 Overview

Kinetic shading devices are inherently adaptive, ideal in the changing environmental conditions throughout the year as shown in Fig. 4. Sensors and internal and external mechanisms allow for kinetic shading devices to continuously adapt to environmental conditions and block unwanted sunlight. A bright day in the summer would inherently create a different shading strategy from an

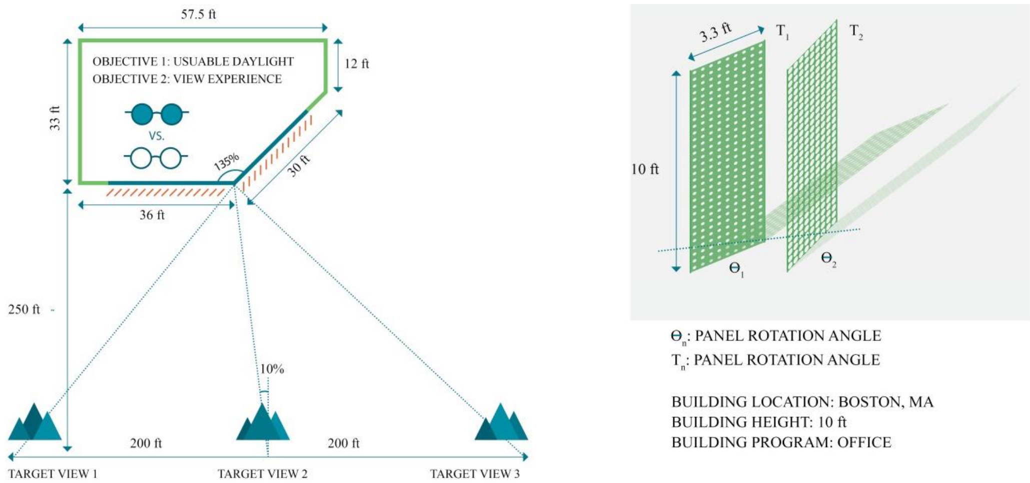


Fig. 6 Problems and design parameters of Case-1

overcast grey day. However, many choose to instead use fixed solutions due to the high cost of installation and maintenance (Reinhart 2001). Fixed shading devices lack the intricate moving parts that require consistent maintenance and overall issue of operability and malfunction. Though attractive in that respect, they pose a problem to the internal users of the designed space. Fixed shades which most often lack of a certain porosity, creates a barrier between a building occupant and the environment beyond the buildings envelope. Though a shading device might adequately block unwanted sunlight, if it fails to allow an occupant to connect with a view, it fundamentally fails as an extension of a window façade (Fig. 5).

With all these considerations in mind, designing and installing a fixed shading solution that does not continuously adapt to the immediate environmental conditions, but is optimized to a certain set of parameters could help to bridge the efficacy and cost of kinetic and fixed shading devices. As such, it is necessary to find a way to resolve the inherent conflict between creating a shading device that works like a kinetic shading device without the drawbacks of a kinetic system. The fixed shading device needs to 1. Perform adequately in minimizing unwanted direct sunlight and 2. Minimize blocking the building occupant's view to the outside environment.

4.1.2 Problem considered and workflow

The problem considered in this study is a hypothetical building of the southern facade is fully glazed with fixed shading devices that can possibly block views to exterior. As shown in Fig. 6, a building is trapezoidal shape and has transparent material at South and opaque material at other sides. The building program is assumed to be a typical HVAC ventilated office building with occupancy hours between 9am and 5pm. Indoor comfort determined by the Percentage of Time Comfortable is calculated using the Predicted Mean Vote, the predicted thermal comfort for the average hypothetical human given a set of climate conditions. ASHRAE standards of 68.5F to 75F in summer months and 75F to 80F in the winter months, both at 50% relative humidity and typical clothing levels and metabolic rates from sitting are used as input conditions for which the occupant is comfortable (Bazjanac 2008). Shading devices of 3-meter high and 1-meter wide perforated metal panel will have varying angle and opacity.

Target view is located in front of the glazed facade in the distance, directly conflicting with the

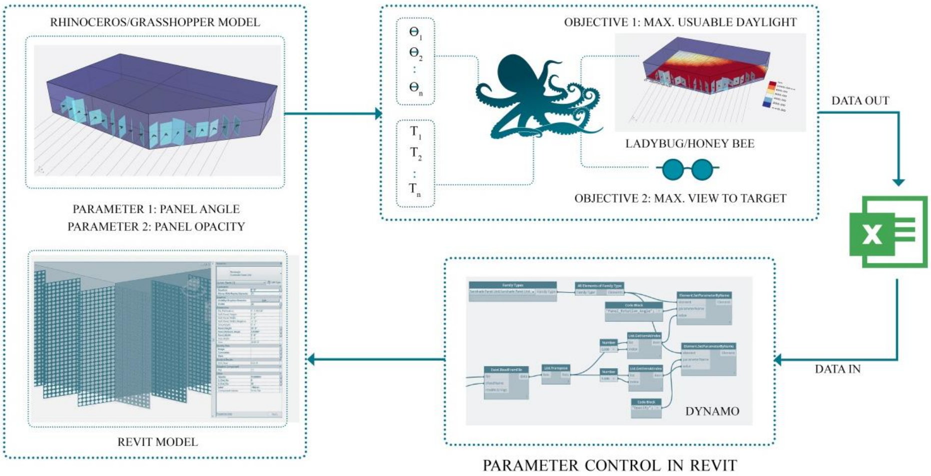


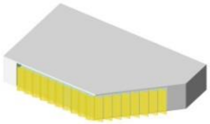
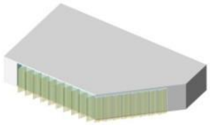
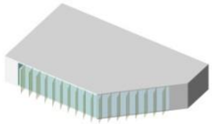
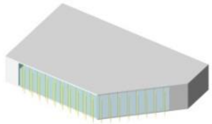
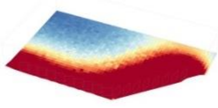
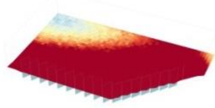
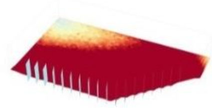
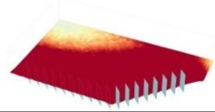
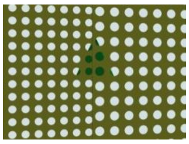
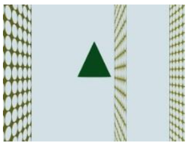
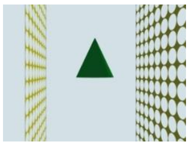
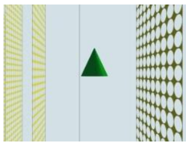
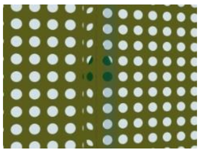
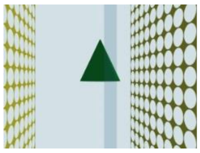
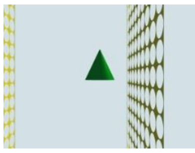
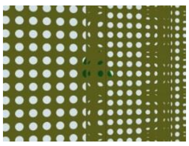
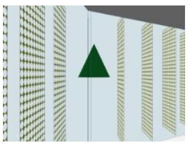
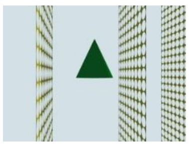
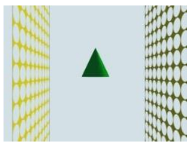
Fig. 7 Workflow Diagram of Case-1

amount of passive solar gains for this study. To test these, the study was set to include three different target views: Target View 1, Target View 2, and Target View 3 as represented in Fig. 6.

As shown in Fig. 7, our workflow employs two modes of information transfer between BIM (Building Information Model) and energy model. The first mode relies on the content of the Revit model elements and parameters to create the input geometry for the energy model in Ladybug and Honeybee/EnergyPlus. In Revit, we create a building and shading device using primitive geometry. Using the shading devices' angle and opacity as instance parameters in a family, the input data creates an individually adaptive family. The primitive geometry is then exported in Mesh to Rhino and using Grasshopper's visual programming converted into Honeybee/EnergyPlus geometry definitions: Zones. Honeybee Zones contain the schedule and internal gain criteria necessary to represent thermal zones, and construction surfaces of Floors, Ceilings, Walls, and Windows. The specific Zones contain material definitions and adjacency relationship to other interior Zones and to the exterior, which thereby form the boundary conditions necessary for Honeybee/EnergyPlus to define for its heat balance algorithms (Garcia 2015).

Given that one point of interest is to understand the relationship between the viewer's experience and the buildings energy use in relation to its passive heat gain, we set the energy model to calculate the number of heating and cooling days required for the building geometry in Boston, a heating dominant climate that requires a particular shading strategy. We use Octopus, a multi-variable evolutionary optimization algorithm, to minimize the total number of heating and cooling days while maximizing the building occupant's view per case analyzed above. Heating and cooling days were used as a function to evaluate when a certain amount of passive daylight would be "useful" and "harmful," further developing a concept of Useable Daylight for this case. For the GA's gene pool, we assign the shading devices angle from 0 to 180 degrees in 10-degree increments while constraining the panel's opacity between 30 and 70 percent, restricted by the structural integrity of a porous panel, in 10 percent increments, limiting an exhaustive simulation. The results are then extracted through a custom data write script through excel and then imported back into Dynamo to define the individual panels in Revit. The consistent BIM data assured by this workflow provides a natural framework for incorporating evolutionary optimization results back into the Revit model.

Table 1 Comparative visualization of Optimization results showing daylight and view experience

Case	Sun only	View 1 Only	View 2 Only	View 3 Only
Exterior View				
Useable Daylight Index				
View from Position A				
View from Position B				
View from Position C				

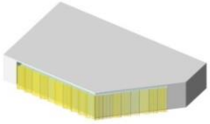
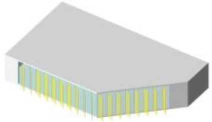
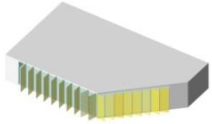
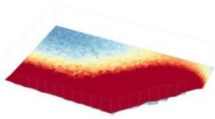
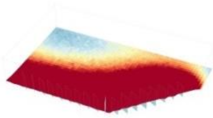
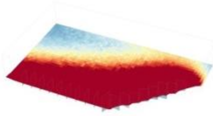
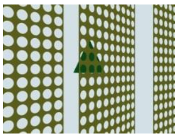
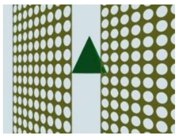
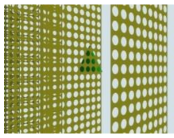
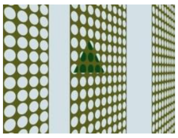
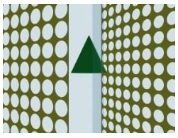
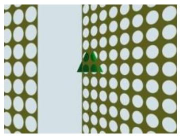
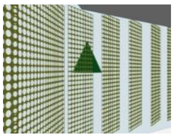
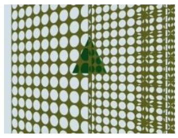
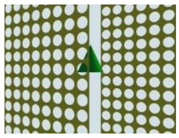
4.1.3 Result and analysis

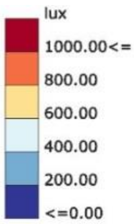
Using Boston as our test climate and 3 different Target Views, we are able to create 3 different optimization scenarios and compare them against single variable optimizations shown in Appendix 1. The single variable optimization scenarios-view and daylight-show the conflicting nature of the objectives. The results show that the genetic algorithms can be used to design individual shading device angles and opacities to optimize and balance building's energy and viewer's experience across unique conditions. Table 1 illustrates the seven different scenarios run with MOF. In order to adequately compare the different scenarios and test our parameters within the MOF workflow, Case-Sun only is the first run with a benchmark single objective optimization utilizing the Sun as the only parameter. The results show that optimizing for the sun only blocks views from all positions A, B, and C. Next, we run the simulation to optimize for the three different views only, which showed how inadequate the passive daylight performance is in all different cases. Finally, to illustrate the strength of MOF and utilizing multiple variables, adequate daylight levels and optimal views to each designated viewpoint V1, V2, and V3. The results of the three multi-variable simulations show shading angles that do not obstruct each viewpoint as well as showing adequate levels of daylight permissivity within the space.

4.2 Case study-2: Interior illumination vs. construction cost

4.2.1 Overview

Table 1 Continued

Case	Sun + V1	Sun + V2	Sun + V3
Exterior View			
UDI			
View from Position A			
View from Position B			
View from Position C			



After successfully optimizing Case 1 which looks to design the ideal space for the user's view experience and shading the space when necessary, we look to adapt MOF to other parameters that designers face in practice. A common passive lighting technique used in large open spaces are large skylights that filter light in for both passive daylighting and heating during heating dominant times of the year. In order to address this, there needs to be an individualized approach to skylights depending on the program needs and the existing environment as noted in Fig. 8, which best practices fail to address. Current skylight best practices include orienting the glazing to the north to reduce over-exposure at certain times of the day A skylight should instead take the total needs of a space at a more holistic, annual approach as illustrated in Fig. 9. Assuming that a skylight system optimized for daylight would adequately suffice the programmatic needs of a space, a significant concern with individualized skylight optimization that needs to be considered also includes a system's high capital cost. Skylights often are static modular systems to reduce fabrication and maintenance costs, and an individualized system requiring multiple fabrication process. With all these considerations in mind, the question then poses: how do you optimally design a low cost, high performance, project specific skylight system? To reduce the total cost of the system, two main factors that remain constant are the number of modular variations and the total material cost of the module itself. Fabrication costs would be lowest if all skylights were the same module and the total area of skylight, thus material cost of the skylights, are minimized. As shown in Fig. 10, if two different skylight scenarios provided the same daylight performance, the lower material cost will be rewarded. Thus, this case study looked to try and quantify an optimal

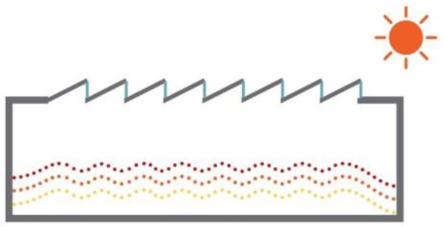


Fig. 8 Different Daylight level requirement per building program

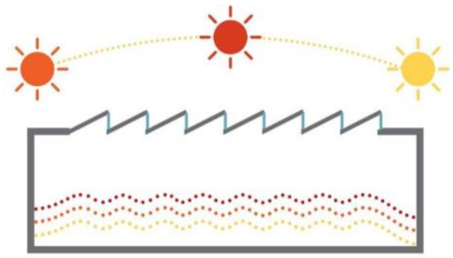


Fig. 9 Different Daylight level throughout a day/a year

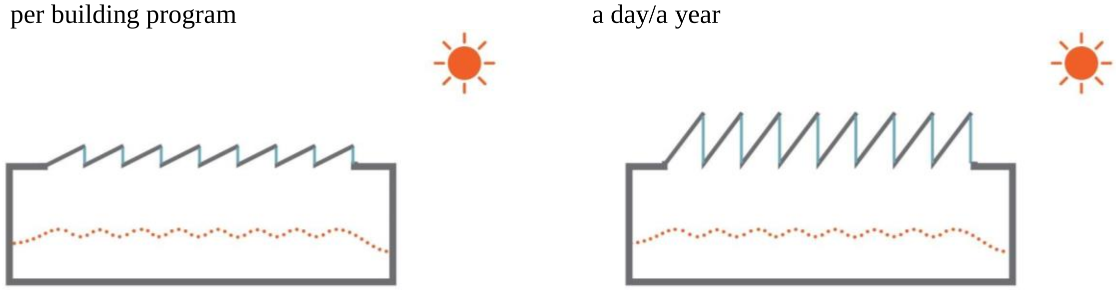


Fig. 10 Different material amount of skylight device for the same daylight performance

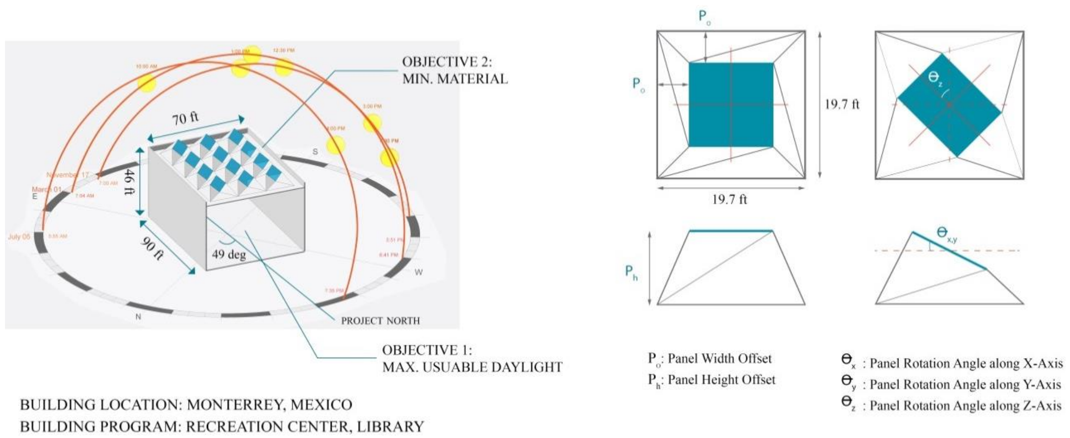


Fig. 11 Problems and design parameters of Case-2

solution for balancing useable daylight with the overall construction cost based on material quantity using MOF.

4.2.2 Problem considered

The problem considered in this study is a hypothetical building of which roof has a cluster of fixed skylights located at Monterrey, Mexico. As shown in Fig. 11, a building is 70’ (W)×90’ (L)×49’ (H) of a rectangular box rotated by 49 degree from the true north. A unit of skylight is approximately 20’×20’ that can fit into roof structure grid.

There are many ways to approach Daylighting Metrics. Often, many will choose 3 point-in-time simulations to illustrate a quick and approximate approach to understanding sun patterns throughout the year. Another approach is to use Annual calculations, which takes every hour of the

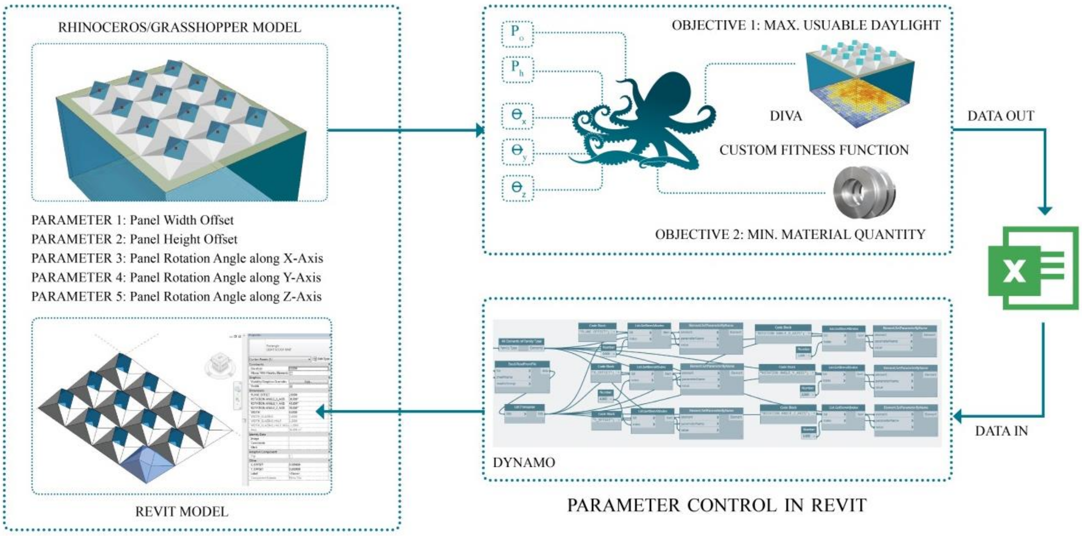


Fig. 12 Workflow diagram of Case-2

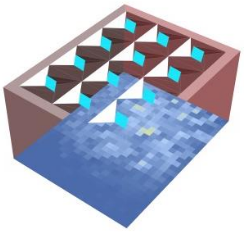
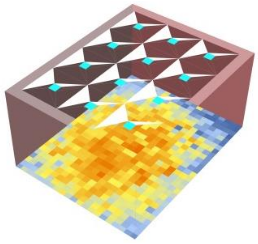
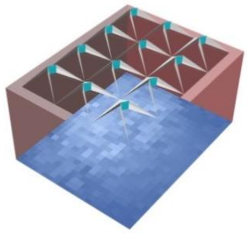
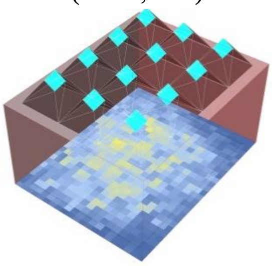
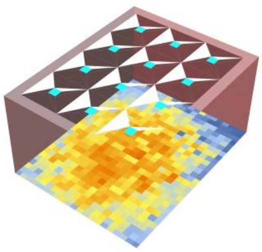
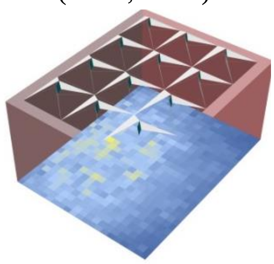
year and presents a more complete picture of total number of day lit hours throughout the year. Given the computational capabilities accessible through the cloud in addition to our desire to most accurately model daylight conditions, we chose to use the annual calculations. Within the realm of annual daylight metrics, there are a few different ways to quantify daylight. The most common metrics used to quantify daylight is Daylight Autonomy. Daylight Autonomy, or DA, is the percentage of the time-in-use that a certain user-designed lux threshold is reached throughout the use of just daylight. The main advantage of DA is that it can be used to assess the quantitative daylight performance of a design-in contrast to a point-in-time calculation that only deals with the worst case scenario. The limitation of DA is that it is solely a quantitative measure. DA is by definition incremental and does not give partial credit for daylight levels below the user-defined lux. Continuous Daylight Autonomy (cDA) is similar to DA, but unlike DA, cDA awards partial credit for daylight levels below a user-defined threshold in a linear fashion.

Like DA, cDA only awards daylight that hits a minimum of the user-defined lux level which fails to address the possibility of over-lighting a space. Useful Daylight Illuminance (UDI) is a modification of Daylight Autonomy, which tries to address the potential glare or overheating defined as beyond 2,000 lux. Like DA and cDA, the user can define the cutoff lux level, but it starts to introduce the concept of over exposing a space to daylight and create parameters for adequate light levels.

For our research, we choose to highlight useable daylight, adequate light without permitting over exposure, so choose to use the metric UDI. In the different simulations, we choose 2 different building typologies, Athletic and Library, to highlight how design changes based on different lux levels. Because we are looking to define useable daylight, we also set our occupied hours to 8am to 8 pm all year round. A library space is defined as requiring 300-500 lux, whereas the Athletic Gym is defined as requiring 200-300 lux. In order to address building costs, we choose a simplified metric of overall material area to calculate total material cost.

In this case study, we start our modeling process in Rhino using parameters created in Grasshopper. We have five parameters to define the geometry of skylight unit that will result in

Table 2 Comparative visualization of Optimization results showing daylight and view experience

Case 1-Sports/Recreation center			
	Case 1-Base/UDI only	Case 1-Base /Min. Material	Case 1-UDI +Min. Material
Parameters			
(θ_x, θ_y ,θ_z)	(50^o, -90^o, -25^o)	(-50^o, 35^o, 65^o)	(45^o, -80^o, -70^o)
(P_o, P_h)	(2 m, 3 m)	(1.5 m, 0 m)	(1.5 m, 2.4 m)
UDI			
(% time)	14.30%	9.52%	12.38%
Area of material (m^2)	664 m^2	438 m^2	594 m^2
Case 2-Library			
	Case 2-Base/UDI only	Case 2-Base / Min. Material	Case 2-UDI +Min. Material
Parameters			
(θ_x, θ_y ,θ_z)	(-10^o, -20^o, -10^o)	(-50^o, 35^o, 65^o)	(20^o, -80^o, 5^o)
(P_o, P_h)	(2.5 m, 3 m)	(1.5 m, 0 m)	(1.5 m, 2.4 m)
UDI			
(% time)	19.57%	13.70%	16.17%
Area of material (m^2)	689 m^2	438 m^2	585 m^2

different daylight and amount of material as shown in Fig. 11:

1. Panel Width Offset
2. Panel Height Offset
3. Panel Rotation Angle along X-Axis
4. Panel Rotation Angle along Y-Axis
5. Panel Rotation Angle along Z-Axis

We again use Octopus utilizing a custom fitness function to evaluate the maximum Useable Daylight and minimum total material quantity. Results are verified by the higher total UDI metric and the lower total material area each compared to the base studies that optimized one variable only. The final data which are notated across the five different parameters above are exported from Grasshopper and imported into Revit which contains the parameters of the original Rhino/Grasshopper Model. This workflow shown in Fig. 12 captures and transfers the data from Grasshopper seamlessly into Revit, showing another way to optimize using MOF with more than one variable.

4.2.3 Result and analysis

Table 2 illustrates the six different Case 2 scenarios run with MOF. In order to adequately compare the different scenarios and test our parameters within this MOF workflow, Case 2 is first run with a benchmark single objective optimization utilizing the UDI as the only parameter for two different programs with corresponding lux level ranges. The results show that maximization for UDI only created a wide range of total material area. Next, we ran the simulation to minimize total material area, which shows how inadequate the total UDI number is in all different cases. Finally, to illustrate the optimization process of MOF's ability to utilize multiple variables, we considered maximizing UDI and minimizing material costs for the two different programmatic daylight levels defined by lux requirements. The results of the two multi-variable simulations show UDI numbers that provide adequate levels of daylight within the space and low total material area.

5. Conclusion and future research

Compared to the typical optimization method in high-performance architecture to seek a design target that relies on single parameter, the Multi-objective Optimization Framework (MOF) proposed in this paper is a BIM-integrated and simulation-based parametric workflow to optimize conflicting multiple variables. In particular, MOF excels in working with performance-based measure and non-performance driven design targets that would provide a direct conflict in the design process, which make it more difficult to satisfy multiple constraints simultaneously in architectural practice.

The MOF integrates the project documentation capability in BIM and the rapid simulation in Rhino, Grasshopper and other plugins, which maximize the benefits of each platform. Two case studies show how the MOF can generate and evaluate all different possible configurations within the predefined each parameter while presenting the most optimized solution and feeding it to BIM environment to reduce lost time and model information. The significance of this research lies in that design practice can be more scientific and automatic by integrating advanced simulation, optimizing algorithm and BIM rather than the exhaustive trial and errors based on a designer's experience and intuition. In this research, the multiple parameters have the same weight, which does not address the current practice where a designer gives certain parameters precedence. Future research can investigate the optimization method with the weighted value per design intents and the user-friendly visualization method.

Acknowledgments

The research described in this paper was financially supported by the Technical Resources Group and the Sustainability Research Group at Sasaki Associates. Thank you to all principals in

charge who have supported our ambitions to bring academic rigor to practice and push our literacy in design and emerging technology.

References

Ali, S. (2010), "Analysis of procedures and workflow for conducting energy analysis using autodesk revit, gbXML and trave 700", *IBPSA-USA J*., **4**(1), 56-63.

Bazjanac, V. (2008), *IFC BIM-Based Methodology for Semi-automated Building Energy Performance Simulation*, Lawrence Berkeley National Laboratory.

Chipperfield, A., Fleming, P., Pohlheim, H. and Fonseca, C. (1994), *Genetic Algorithm Toolbox User's Guide*, Department of Automatic Control and Systems Engineering, University of Sheffield, Version, 1.

Garcia, E.G. and Zhu, Z. (2015), "Interoperability from building design to building energy modeling", *J. Build. Eng*., **1**, 33-41.

Kilian, A. (2006), "Design innovation through constraint modeling", Ph.D. Dissertation, MIT, Cambridge, U.S.A.

Miller, N. (2011), http://wiki.theprovingground.org/usc-arch517-galapagos, The Proving Ground.

Reinhart, C.F. and Walkenhorst, O. (2011), "Validation of dynamic RADIANCE-based daylight simulations for a test office with external blinds", *Energy Build*., **33**(7), 683-697.

Roudsari, M.S., Pak, M. and Smith, A. (2013), "Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally conscious design", *Proceedings of the 13th International IBPSA Conference*, Lyon, France.

Thiele, L., Miettinen, K., Korhonen, P.J. and Molina, J. (2009), "A Preference-based evolutionary algorithm for multi-objective optimization", *Evolut. Comput*., **17**(3), 411-436.

Vierlinger, R. and Hofmann, A. (2013), "A Framework for flexible search and optimization in parametric design", *Proceedings of the Design Modeling Symposium Berlin*.

Welch, R., Egerter, A., Tucker, S. and Connock, C. (2016), "A BIM workflow for iterative and informative energy modeling", *Proceedings of the ASHRAE and IBPSA-USA 2016 Building Performance Modeling Conference*, UT.

Yi, Y.K. and Malkawi, A.M. (2009), "Optimizing building form for energy performance based on hierarchical geometry relation", *Automat. Constr*., **18**(6), 825-833.

York, D.A. and Cappiello, C.C. (1981), *DOE-2 Engineers Manual (Version 2. 1A) (No. LBL-11353; LA-8520-M)*, Lawrence Berkeley Lab., CA, U.S.A., Los Alamos National Lab., NM, U.S.A.

Zitzler, E., Laumanns, M, and Thiele, L. (2001), *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Swiss Federal Institute of Technology (ETH) Zürich, TIK-Report 103, 2001.

TK

Appendix: Angle and opacity of optimized shading panels at case study 1

	Sun		V1 Only		V2 Only		V3 Only		Sun + V1		Sun + V2		Sun + V3	
	θ	T (%)	θ	T (%)	θ	T (%)	θ	T (%)	θ	T (%)	θ	T (%)	θ	T (%)
P1	20	70	60	30	-80	30	-40	30	20	60	-40	50	90	60
P2	20	70	60	30	-80	30	-40	30	20	60	-50	50	90	60
P3	20	70	60	30	-80	30	-40	30	10	60	-30	50	90	60
P4	20	70	60	30	-70	30	-40	30	20	60	-30	50	90	60
P5	20	70	60	30	-70	30	-40	30	10	50	-30	50	90	60
P6	20	70	60	30	-70	30	-40	30	10	60	-30	50	90	60
P7	20	70	60	30	-70	30	-40	30	20	60	-30	50	90	60
P8	20	70	60	30	-70	30	-40	30	20	50	-40	40	90	60
P9	20	70	60	30	-70	30	-40	30	20	50	-50	50	70	40
P10	20	70	60	30	-70	30	-40	30	20	40	-50	50	70	50
P11	-30	70	10	30	60	30	90	30	0	50	-70	40	-50	50
P12	-30	70	10	30	60	30	90	30	0	40	-70	40	-30	40
P13	-30	70	10	30	60	30	90	30	-10	50	-70	40	-20	60
P14	-30	70	10	30	60	30	90	30	-10	40	-70	40	-30	50
P15	-30	70	10	30	60	30	90	30	-10	30	-70	40	-30	50
P16	-30	70	10	30	60	30	90	30	-20	40	-70	50	-40	40
P17	-30	70	10	30	60	30	90	30	-10	50	-80	60	-40	60
P18	-30	70	10	30	60	30	90	30	-10	50	-80	60	-20	60