Journal of Building Performance Simulation
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tbps20

A generative facade design method based on daylighting performance goals

Jaime Gagne [a] & Marilyne Andersen [a]
[a] Department of Architecture, Massachusetts Institute of Technology, Building Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139-4307, USA
Published online: 30 Mar 2011.

Journal of Building Performance Simulation
Vol. 5, No. 3, May 2012, 141–154

A generative facade design method based on daylighting performance goals

Jaime Gagne* and Marilyne Andersen

Department of Architecture, Massachusetts Institute of Technology, Building Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA

(*Received 21 June 2010; final version received 15 December 2010*)

Successful daylighting design is a complex task which requires the designer to consider numerous design elements and their effects on multiple performance criteria. Facades, in particular, include many variables which may dramatically impact daylighting performance. Genetic algorithms (GAs) are optimization methods which are suitable for searching large solution spaces, such as those presented by design problems. This article presents a GA-based tool which facilitates the exploration of facade designs generated based on illuminance and/or glare objectives. The method allows the user to input an original 3d massing model and performance goals. The overall building form remains the same while facade elements may change. Ten parameters are considered, including materials and geometry of apertures and shading devices. A simple building data model is used to automatically generate a 3d model of each solution. Results from single- and multi-objective case studies are presented to demonstrate a successful goal-driven design exploration process.

Keywords: daylighting; genetic algorithm; facade optimization; generative design system

1. Introduction

The facade design of a building is possibly the most critical element in creating a successful daylighting scheme on the interior. Optimization algorithms, such as genetic algorithms (GAs), have the potential to aid in performance-based facade design by combining an intelligent search process with performance output from simulation engines. This article presents a GA-based method for facade design exploration, which can be integrated into the design process. The proposed method considers both illuminance and glare metrics to enable a complete understanding of daylighting performance due to facade elements. To appeal to designers, the method has been implemented in SketchUp (Google 2010), an intuitive 3d modelling environment. Within this environment, the final solution is generated as a 3d model or set of models that the designer can use as a starting point as he or she continues the design process. The method represents a first step towards integrating performance-based search into the early design exploration process.

Numerous studies have already demonstrated the potential for optimization algorithms to facilitate performance-based facade design exploration. Several researchers have considered photovoltaic-integrated facade systems and examined the trade-off between facade area used for daylighting and that used for electricity generation (Vartiainen et al. 2000, Charron and Athienitis 2006). Park et al. considered double facade systems with integrated blinds and have found optimal blind angles for several visual comfort metrics (Park et al. 2004). Shea et al. (2006) optimized the effect of the glazing type of roof panels on lighting performance and cost. Several studies have optimized window size and placement while considering both daylighting and energy (Caldas and Norford 2002, Wright and Mourshed 2009). In the GENE_ARCH system, lighting and energy are optimized in a generative system which can also incorporate an architect's specific aesthetic design intent (Caldas 2008). Other studies have considered daylighting performance from a visual comfort standpoint. For example, Chutarat's system allowed multiple objectives within the daylighting domain such as illuminance, glare and direct sunlight (Chutarat 2001), and Torres and Sakamoto's study found facade solutions resulting in high illuminance and minimal glare due to daylighting (Torres and Sakamoto 2007).

Although there have been many previous studies which focused on facade optimization, few have been suitable to be implemented into an actual design process. Many of these studies have restricted the scope of the problem by fixing the initial geometry of the space and the main optimization objective (typically minimizing energy consumption due to electric lighting). Such limitations are very restrictive in an

*Corresponding author. Email: jaimelee@mit.edu

ISSN 1940-1493 print/ISSN 1940-1507 online

http://dx.doi.org/10.1080/19401493.2010.549572
http://www.tandfonline.com

actual design scenario, as users may not be able to model a problem that is relevant to their specific design goals and aesthetics. A designer who might want to adapt an existing optimization process to meet his own specifications would either need to interpret the provided output to his specific problem or design, or possess knowledge of programming languages and appropriately modify the algorithm himself.

The proposed approach aims to incorporate performance-based daylighting design exploration and optimization into the design process by offering a GA-based method which can be customized to suit a designer's specific needs without requiring skills beyond 3d modelling. A simple building data model has been created to allow the user-defined massing model to be understood by the system. Object-oriented and building data model approaches such as building information models (BIMs) have been proposed in the past as a way to integrate optimization into the design process by allowing designers to use optimization in familiar computer-aided design (CAD) based settings and to enable a more generic approach which can handle a larger variety of problems (Wang et al. 2005, Geyer 2009). The simple building data model described in this article enables the system to recognize the geometric characteristics of the user's initial model and to automatically generate new 3d models during the GA process. The user does not need to create the building data model; instead, a custom data model is automatically created and populated based on the user's original 3d model at the beginning of the process.

This article presents a validation of the proposed system using simple case studies and an application of the system to three more complex design scenarios. These problems include a single-objective problem with two non-conflicting goals and two multi-objective problems. One of the multi-objective studies includes two conflicting illuminance goals, and the other one deals with conflicting illuminance and glare goals. In all the situations presented, the proposed method was able to successfully explore the design space and present the user with a design solution or set of solutions which approach the user-defined performance objectives. These generated models are good starting points for a designer who is interested in considering daylighting performance in the early design stages.

2. Proposed approach

The system described in this article has been created using Google SketchUp's embedded Ruby application programming interface (API). This section will describe the various components of the system, including the daylighting simulation engine and metrics, the optimization problem and algorithm, the required user inputs and the automation of generated 3d models using a simple building data model.

2.1. *Daylighting simulation engine*

A building optimization process requires computationally intensive simulations. Because the proposed system is intended for use during the design process, an efficient simulation engine is a necessity. The engine used in the proposed approach, the Lightsolve Viewer (LSV), is a hybrid global rendering method which combines forward ray tracing with radiosity and shadow volumes rendering (Cutler et al. 2008). This engine was chosen because it allows for rapid calculation of the daylighting metrics described in the following section. Cutler et al. found that a rendered scene in LSV took approximately 3.3% of the time that it took to complete an analogous ''fast rendering'' in radiance. Early validation results indicated that rendered images by LSV displayed a pixel difference of less than 10% from radiance for a variety of scenes, camera positions and daylighting conditions (Cutler et al. 2008).

To make the whole-year simulation more efficient, the LSV engine divides the year into 56 periods and calculates the illuminance during each time period under four different sky types ranging from overcast to clear using the method described in Kleindienst et al.'s paper (2008). The climate-based illuminance is then calculated for each time period as a weighted average of illuminances from each sky type. In this study, the total computation time for a full-year simulation with illuminance and glare results ranged from less than 1 min for a simple model to about 5 min for a more complex model on the author's computer.[1] An analysis comparing illuminance data calculated on point sensors in radiance with area-based patch sensors in LSV indicated similar values (5% median, 7% mean and 28% maximum relative difference) for a model similar to those considered in the present study (Lee et al. 2009).

The LSV engine currently models glazing materials as ''virtual'' glass, where transmittance is independent of the solar incidence angle. In the future, more materials, including realistic glass, interior shading, and advanced fenestration materials, will be available.

2.2. *Optimization problem*

Because the desired daylighting conditions may differ among various design scenarios, the proposed system features metrics which are calculated based on goals specifically inputted by a user. To allow users to understand the daylighting performance of their space in terms of both quantities of light and visual comfort, we considered two metrics, one for illuminance and one for daylight-based glare. The design variables, chosen for this study, are those which are known to influence both of these metrics and which are generally considered early in the design process. This section describes the design goals and design variables in further detail.

2.2.1. Daylighting metrics and design goals

To allow for a comprehensive understanding of daylighting performance, the proposed approach features two different types of daylighting metrics, one for illuminance levels and one for glare. Both of these metrics are calculated using the daylighting simulation engine described in section 2.1. The goal-based illuminance metric requires the user to input four illuminance values: acceptable low, desired low, desired high and acceptable high (Kleindienst 2010). The user must also specify which time periods of day and seasons he or she is interested in: morning, mid-day, afternoon and winter, spring/autumn, summer. This metric is derived from the work presented by Kleindienst et al. (2008) and uses the same logic for climate and temporal simplifications. It assumes a user-defined sensor plane which will be divided into small patches during the simulation process. For a single patch, the goal-based illuminance metric is defined as the percentage of the user's times and seasons of interest in which daylight provides an illuminance within the user's specified range. The final goal-based illuminance for a sensor plane is an average of the performance over all patches on a sensor plane. For illuminance levels which fall between the ''acceptable'' and ''desired'' values, partial credit is given (Figure 1(a)). A value of 100% indicates that the entire area of the sensor plane sees an illuminance in the user's desired range over all periods of day and seasons of interest.

The glare metric used in the proposed approach is a model-based approximation of daylighting glare probability (DGP), which has been developed by Kleindienst and Andersen (2009). The DGP metric, originally introduced by Wienold and Christoffersen (2006), indicates the percent of occupants disturbed by a daylighting glare situation. The model-based DGP approximation method (DGPm) is an efficient way of approximating the DGP using the LSV engine. When compared to the DGP as calculated using the evalglare program in radiance for different rectangular spaces (and only for windows without mullions), the DGPm has been found to perform within a 10% error over 90% of the time (Kleindienst and Andersen 2009). The metric assumes a user-defined vertical sensor plane whose normal indicates a direction of view. To evaluate glare risks, the present study uses the glare thresholds described by Wienold (2009), where any value below 0.33 (imperceptible glare) is considered a ''no glare'' situation and given a glare credit of 0. The user may choose from three glare tolerances: ''zero'', which corresponds to an upper glare threshold value of 0.37; ''medium'', which corresponds to a threshold value of 0.42; and ''high'', which corresponds to a threshold value of 0.53. Any calculated glare value above the upper threshold is given a glare credit of 1 (Figure 1(b)). These glare credits are averaged across all glare sensors which face the same general direction within the model. A value of 0% indicates that the specified view direction is unlikely to see glare due to daylighting.

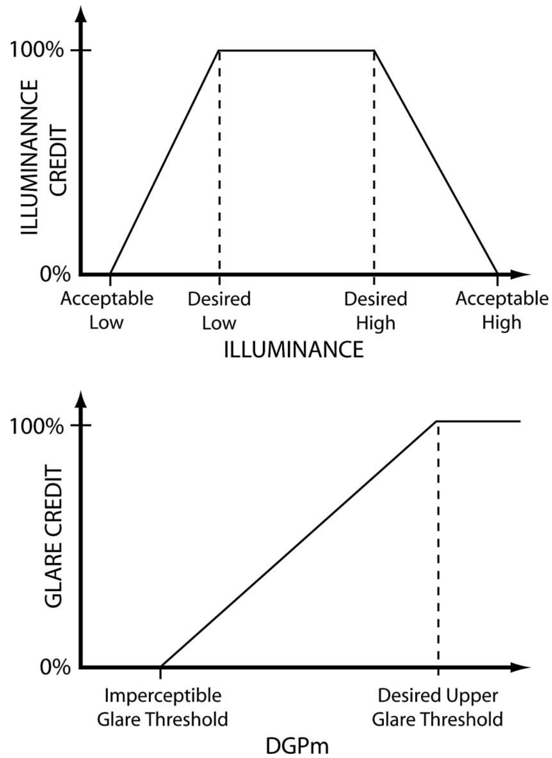


Figure 1. Diagrams indicating system of full and partial credit for (a) illuminance and (b) glare.

Because the daylighting performance metrics are defined as percentages, the objectives for any user-defined problem can be represented as:

– Maximize the percentage of illuminance levels within the user-selected illuminance goal range(s) on the illuminance sensor plane(s); and
– Minimize the percentage of glare (based on the user-selected tolerances) perceived by the glare sensor plane(s).

This formulation allows for the same search algorithm to be used for a wide range of daylighting optimization problems. The scope of the problem includes illuminance and/or glare goals for models which conform to the guidelines described in Appendix.

2.2.2. *Design variables*

Ten different design variables are considered, as indicated in Table 1, along with the minimum and maximum values they can take and the step sizes. For this study, only parameters associated with the facade were used. These parameters were chosen because they are typically considered early in the design process and frequently have a large impact on both exterior aesthetics and on daylighting performance, including both illuminance and glare.

2.3. *Micro-genetic algorithms*

GAs (Goldberg 1989) have been applied to many types of architectural problems. During the GA process, a set of initial solutions (a population) is chosen or generated at random. Each member is evaluated for "fitness" (performance) and members that result in good performance are used as "parents" for a new generation. Since this new generation is based on the best performing solutions in the previous solutions, we assume that some members of the new generation will perform better. Once evaluated, the good performers are used as parents while the poor performers are discarded. The cycle is continued until a suitable solution or set of solutions is found or until a predetermined number of generations have been completed.

Multi-objective GAs work in a similar way except that in these cases, one might consider two or more objectives which are conflicting. In such cases, increasing the fitness of one objective may decrease the fitness of another, which means that a single optimal solution may not exist. Instead, it is traditional to try to find the Pareto front, which is the set of all solutions in the solution space that are non-dominated or Pareto-optimal. If for a given solution, we can find another one within the solution space that is better for both objectives, that solution is considered strongly dominated. Pareto-optimal or non-dominated solutions are those which are not dominated by any others within the solution space. For a more comprehensive explanation of multi-objective problems and Pareto optimality, see Coello Coello et al. (2007).

GAs typically require large population sizes and numbers of generations to converge, particularly for multi-objective problems where the desired result is not a single solution but a set of Pareto-optimal solutions. In this study, we used a micro-GA, a genetic algorithm which uses a very small population size when compared to a classical GA. This small population size reduces the computational time necessary to simulate each generation, which means that a user can run several generations of the micro-GA using the same number of simulations as a single generation of a classical GA. Micro-GAs have also been shown in some studies to require fewer function evaluations than a classical GA to converge to the near-optimal region (Krishnakumar 1989, Carroll 1996). Micro-GAs have been successfully implemented for building performance optimization based on building energy criteria, lighting and thermal behaviour (Caldas and Norford 2002, Caldas 2008). The proposed approach allows for both single- and multi-objective problems, which both utilize a micro-GA algorithm. The single-objective problem considers illuminance only, while the multi-objective problem considers both illuminance and glare.

Table 1. List of variable facade parameters and possible values.

Facade parameter	Minimum value	Maximum value	Step size
Window-to-wall ratio	0.1	0.8	0.1
Number of windows	1	8	1
Aspect ratio[a]	Thinnest	Widest	–
Vertical location[a]	Lower bounds	Upper bounds	–
Horizontal location[a]	Right bounds	Left bounds	–
Window distributions[a]	Windows touching	Windows far apart (at bounds)	–
Overhang	Yes	No	–
Fins	Yes	No	–
Length of shading devices	0.15 m (0.5 ft)	1.22 m (4 ft)	0.15 m (0.5 ft)
Total glass transmissivity (%)	10	85	5
Per cent specular transmission (%)	0	100	12.5

Note: [a]Actual values for these parameters will depend on user-defined geometry.

2.3.1. Single-objective micro-GA

The micro-GA that has been implemented within the proposed system is the original single-objective algorithm as described by Krishnakumar (1989). Encoding is done using binary strings, where the full set of design parameters described in section 2.2.2. is encoded into a string of 30 bits for each separate facade considered. A micro-GA differs from a traditional GA in several ways. The most obvious is the small population size: five individuals in our system. Due to this small population size, a micro-GA population tends to reach "bitwise convergence" within only a few generations. Bitwise convergence is reached when all individual binary strings in the population differ by 5% or less. Upon bitwise convergence, the micro-GA resets the population by creating a new random population. It is of note that this algorithm does not use mutation, as it is assumed that enough diversity will be maintained in the population through the generation of new random strings upon bitwise convergence, which is likely to occur numerous times during the optimization process.

In our system, for the single-objective problem, fitness is defined as the goal-based illuminance for a single sensor plane or the average goal-based illuminance over multiple sensor planes. An "optimal" solution will be one in which fitness is found to be 100%, which indicates that the illuminance goals are met over 100% of the sensor plane area and over 100% of the time during a year. For a simple problem, it may be possible that multiple solutions within the solution space meet the goal criteria (an example of this phenomenon is described in section 3.1.).

Because we intend for the proposed method to be used by designers early in the design stage for design exploration and not necessarily for true optimization, we do not impose a strict stopping criteria, such as a mathematical or algorithmic test for global convergence. The system will stop upon finding a solution which meets the goal criteria, or else the designer can stop the process after a predetermined number of generations have been completed.

2.3.2. Multi-objective micro-GA

The micro-GA has previously been successfully used for multi-objective problems (Coello Coello and Pulido 1993) by including external memory which stores non-dominated solutions generated over the course of the process. For this study, the algorithm used is similar to that described for single-objective problems (Krishnakumar 1989), with the addition of an external memory similar to that described by Coello Coello and Pulido. A binary Pareto fitness ranking is used, and at each step, the memory is updated to include new non-dominated solutions, and any previous solutions which are dominated by new ones are then removed. A pseudo-Pareto front is approximated to be those solutions contained within the external memory after a certain number of generations. The multi-objective process is essentially the same as the single-objective process except it works towards finding non-dominated solutions instead of working towards a single solution with the highest fitness.

It is important to note that while this process does produce a set of non-dominated solutions, which may approximate the Pareto front, it does not necessarily generate a true Pareto front with evenly distributed solutions. However, as this system is intended to be incorporated into the early design stages, the generation of a true Pareto front would likely require a number of simulations that would be too time consuming to complete. Although further research would be needed to confirm this, one might also argue that a true Pareto front may actually not be required for designers who wish only to see a range of possible solutions and who will ultimately be using them as a starting point, not as a final design, as they continue in their design process.

Within the described system, a user would use a multi-objective approach when he or she had two conflicting performance goals, or two sets of conflicting goals. One scenario is two illuminance goals, in which case the objective is to maximize the goal-based illuminance on both sensors. The other scenario is combined illuminance and glare, in which case the first objective is to maximize the goal-based illuminance on all illuminance sensors and the second objective is to minimize the model-approximated DGP on all glare sensors. Similarly to the single objective problem, we do not impose strict stopping criteria for the multi-objective case. Instead we allow the user to choose a predetermined number of generations to run before stopping the process.

2.4. Integration and operation

Within the proposed system, one important way in which we have approached the problem of integrating a GA-based tool into the design exploration process is to allow the user to quickly and intuitively model his or her specific design problem. Instead of specifying the massing design using text-based inputs, the user is allowed to create a 3d model in Google SketchUp as input. This type of input should facilitate use of the proposed system, particularly for designers who would typically create such models over the course of the early design process anyway. Additionally, the system automatically generates and saves 3d models of all solutions found over the course of the GA process.

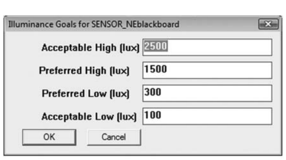
Once the process is complete, the designer can use these models directly to further their designs. This section describes the inputs and outputs of the proposed system, including the use of a simple building data model which is used to automate the process of generating new 3d models.

2.4.1. User inputs

One way in which the proposed method has been developed to appeal to designers is to incorporate an intuitive set of user inputs. In particular, one innovation is to allow the user to specify the base model by creating a 3d massing model in Google SketchUp instead of requiring them to define their base model using a text-based approach. This user-defined massing model should indicate the general form of the space and all desired opaque material properties, i.e. wall, floor, and ceiling reflectances. In order for the system to correctly interpret the massing model, the model should conform to a few basic guidelines (see Appendix). We chose to require the user to name the materials chosen for certain design elements using special names (glazing, shading devices and those facades which would be generated by the GA) rather than to have the user select those elements directly, as we assumed that they would be specifying materials anyway.

Within the 3d model, the user must also specify 2d sensor planes on which either illuminance or glare will be calculated by including these planes in the massing model. The user may choose to have any number of illuminance and/or glare goals. The sensor planes may be any size, and they may be oriented vertically or horizontally. For each illuminance sensor plane, the user must specify a desired illuminance goal range in lux, and for each glare sensor, the user must indicate a desired glare tolerance (see section 2.2.). The user can input these goals into simple interfaces in Google SketchUp (Figure 2).

a.



b.

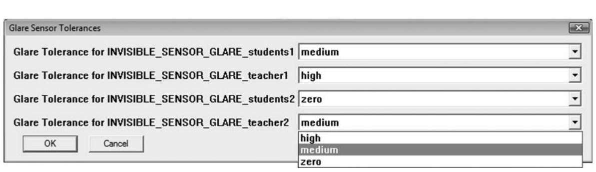


Figure 2. Goal input boxes for (a) illuminance ranges and (b) glare thresholds.

2.4.2. Building data model

One of the novel features of the proposed approach compared to work cited previously is the ability for the user to provide a 3d model as input instead of requiring programming, text-based input or the use of a default model. To provide this functionality, a building data model was developed whose values are automatically assigned once the process is initiated. The model contains information about each building element in a 3d model and the relationships between them. The general structure of the data model is indicated in Figure 3. Each building element object contains information about its location, geometry, orientation and material.

The building data model allows the algorithms in the proposed approach to understand which walls are to be manipulated by the GA and what the boundary conditions of those walls are. It also allows the system to automatically create new 3d models of each GA population member which can then be simulated during the GA process.

In the proposed approach, the user creates a 3d model in SketchUp as an initial input and a simple building data model is automatically created by the system. The logic for this automatic model population is defined in detail in Appendix. Identification of each building element occurs using a series of logic

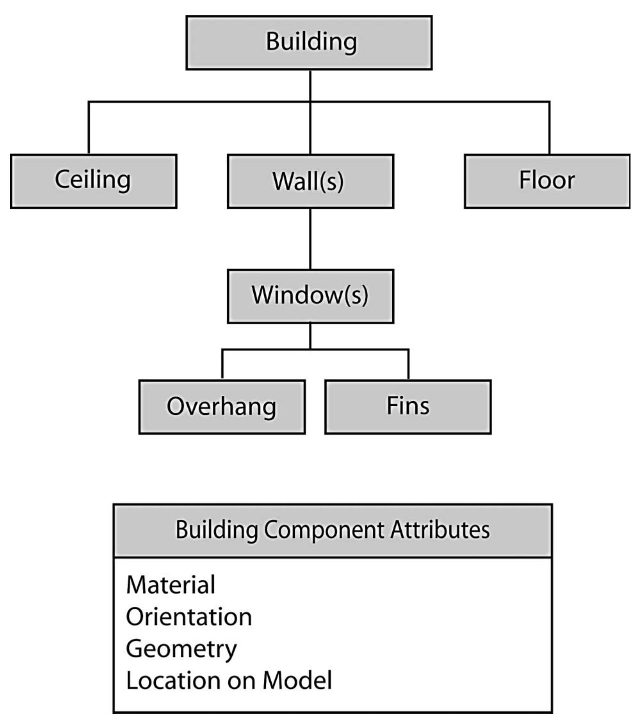


Figure 3. Schematic of simple building data model: relationships between components and object attributes.

statements regarding the geometry and material of each modelled component. Element attributes are then determined using information available from SketchUp about each face. The logic assumes that the model conforms to a few basic guidelines (see Appendix).

The use of the building data model is necessary because SketchUp is a geometric modelling tool and not a BIM. While there do exist plugins for BIMs within SketchUp (for example, Demeter (Greenspace Research 2007)), these programs require the user to input more information than what is necessary for our process and the population of the data model is generally not automated. However, if an appropriate BIM were to become available within SketchUp, it would be possible to integrate data from such a BIM into our system.

2.4.3. Model generation

The proposed approach automatically generates 3d model representations of the binary strings created during the GA process, i.e. it creates and saves new SketchUp models for all population members. These models are created using the following process (Steps 1 through 5 are demonstrated for an example model in Figure 4):

(1) Add a single window of the given window-to-wall ratio (window area) to the facade using the same width as the wall itself to ensure fit.
(2) Divide into the given number of windows.
(3) Calculate the highest and lowest possible aspect ratios that the windows can take based on the window size and wall dimensions. Change the aspect ratio of all windows based on the given value.
(4) Calculate the largest distance that can exist between each window based on window size and wall dimensions (assume smallest distance is 0.05 m [2 inches]). Change distribution based on given value.
(5) Determine upper, lower, left and right wall boundaries. Change window group location based on given value.
(6) Add shading devices of the given length, if applicable.
(7) Change window material given values.

Because the geometric parameters (window aspect ratio, location and distribution) are calculated based on the boundary conditions of a given facade instead of being based on absolute values, the proposed approach can generate models using any type of original massing geometry that features vertical walls facing cardinal directions. The user can also choose to rotate the sky so as to simulate models whose walls are orthogonal but which are not aligned with the cardinal axes. This feature provides the user with a great level of flexibility when creating the original massing model.

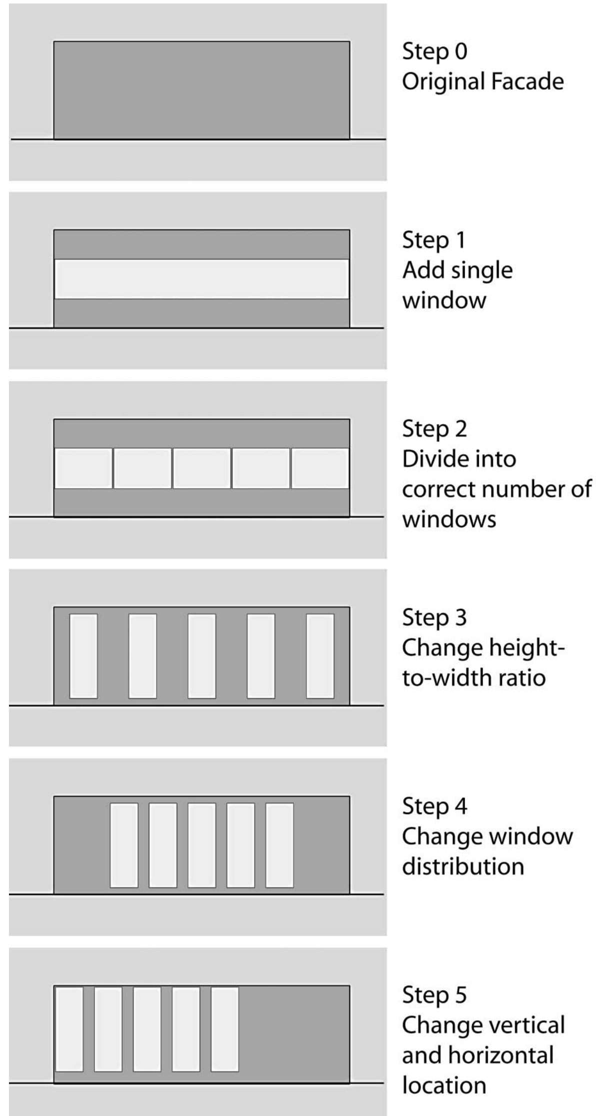


Figure 4. Automatic facade generation process based on geometry variables.

3. Validation

To ensure that the micro-GA algorithm was behaving as expected, a set of test studies were performed on a simple box model with a single illuminance sensor plane located in the centre of the space at workplane height. For each of these studies, the south and east facades were generated by the GA while the north and west facades remained opaque. In both cases, one or more solutions to the problem were known to exist and had been manually found by the authors before the case studies were conducted. To determine the general behaviour of the algorithm, for each case study the GA process was run three times over ten generations or until a ‘‘perfect’’ solution was found. Because we wanted to verify that the algorithm would converge quickly, we chose ten generations as stopping criteria as it seemed a likely number that a designer might consider choosing for a simple design scenario. For these studies, ten generations took roughly 1 h to simulate on the author’s computer.

Both case studies were considered successful, although some limitations to the GA method can already been seen in these initial trials. These limitations include inconsistencies in the number of generations required to find a good solution, which are due to the random and probabilistic nature of the algorithm, and the possibility for the algorithm to get ‘‘stuck’’ in one part of the solution space, which may be due to the lack of mutation in the algorithm or to the nature of the binary encoding that was used. However, these studies also demonstrated the potential for the micro-GA to effectively search a broad design space and to converge onto successful designs quickly. While some limitations were observed, these behaviours are common to many types of optimization algorithms, and the GA is known to be less likely to fall into local minima than some other algorithms.

3.1. *Illuminance goal with no minimum*

The maximum illuminance values in this study were 200 lux (desired) and 400 lux (acceptable); no minimum values were specified. All seasons and periods of day were considered. This case study was considered the most basic because several designs within the search space were known to meet the desired illuminance goal range. The known solutions featured a small window area with long shading devices. As expected, the micro-GA was highly successful at determining solutions to this problem, finding a ‘‘perfect’’ solution on each of three separate runs, each time within 10 generations. For each of the three runs, the performance of the GA and the final generated solution are indicated in Figures 5 and 6, respectively. Because the solution space was known to be highly multi-modal, it is not surprising that the three solutions found all met the goal criteria yet all had different forms. It is likely that the final solutions are different because each run began with a different random initial population, although the probabilistic nature of algorithm contributes to this diversity as well. It is possible that additional runs would have produced even more varied results, although it is likely that any additional solutions would feature similar characteristics of small window area with large shading devices.

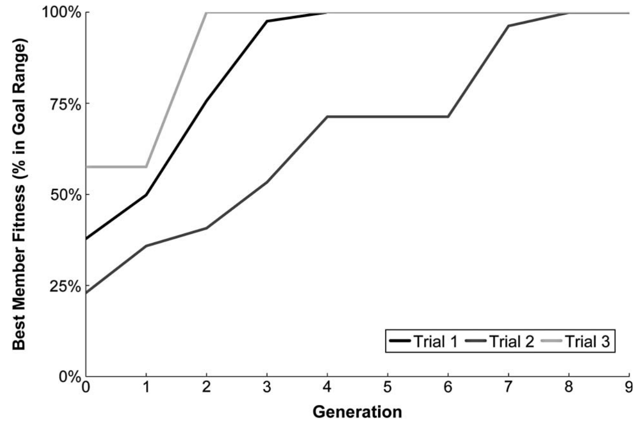


Figure 5. Best member fitness over 10 generations for case study with no minimum illuminance (three trials).

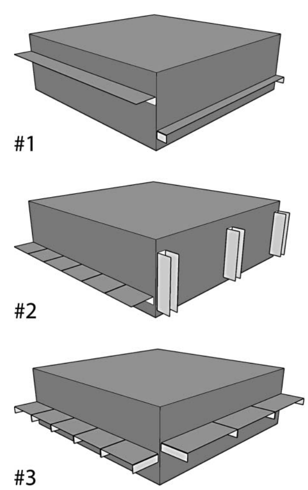


Figure 6. Final solutions for each of three trials for case study with no minimum illuminance.

3.2. *Illuminance goal with no maximum*

This case study used the same model and sensor plane as the previous study, but the goal in this problem was to obtain a desired 400 lux minimum (200 lux minimum accepted) with no maximum values. For this problem, the authors were only able to manually find one ‘‘perfect’’ solution, although it is possible that more solutions exist. In this case, the known solution featured a large window with a high glass transmissivity and no shading devices. This case study was considered more difficult than the previous one because there were fewer known solutions within the search space.

As expected, the micro-GA came very close to finding the known solution after ten generations but never found one in which the illuminance goals were met over 100% of the sensor plane area and over the whole year. The most successful of the three final solutions generated has the same features as the known solution. All three solutions had a very large window area; however, it is interesting to note that the second solution got "stuck" in a search space that only included shaded designs, so the final solution in this case is the worst performer of the three. It is likely that this solution would have become "unstuck" if we had allowed the algorithm to continue running for more generations, which would have introduced new random solutions into the population upon convergence, or if we had modified the algorithm to include mutation. Nevertheless, all three trials found good designs after only a few generations, indicating that the micro-GA was again successful at efficiently converging on good designs (Figures 7 and 8).

4. Application case studies

While the validation case studies may be considered successful, those studies represented only the most simple performance goals and would likely not be applicable to a true design scenario. In this section, we describe three more complex case studies. All three case studies have non-rectangular footprints and multiple goals. The first case study has two non-conflicting illuminance goals, the second has two conflicting illuminance goals, and the third has conflicting illuminance and glare goals. These studies represent the variety of designs and performance goals that can successfully be explored using the proposed system.

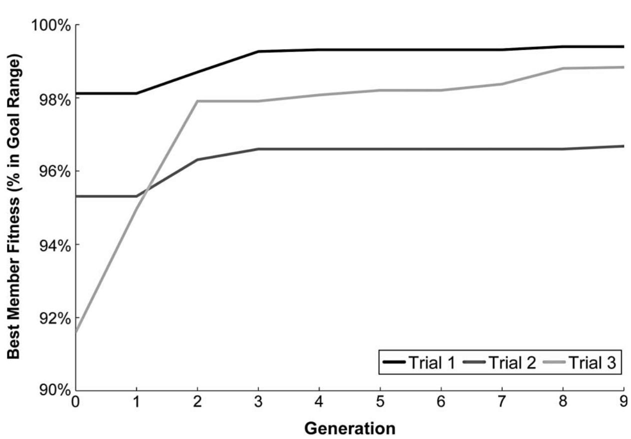


Figure 7. Best member fitness over 10 generations for case study with no maximum illuminance (three trials).

4.1. *Case study no. 1: non-conflicting illuminance goals*

The proposed GA approach was applied to the massing model shown in Figure 9 in Boston, MA. This model has a non-rectangular footprint and a slanted roof condition. The facades of interest in this model were those facing north and south. It has two illuminance goals that were not considered conflicting. Both sensor planes are located at workplane height. The illuminance goals for the west sensor are 200 lux (acceptable) and 400 lux (desired) lower bounds; no maximum. The goals for the east sensor are 100 lux (acceptable) and 200 lux (desired) minimum; 800 lux (desired) and 1000 lux (acceptable) maximum.

The micro-GA process was run for a total of 25 generations. We chose 25 generations as a reasonable number for a designer working on a somewhat complex problem. In total, the simulations required for 25 generations took approximately 4 h, or half a

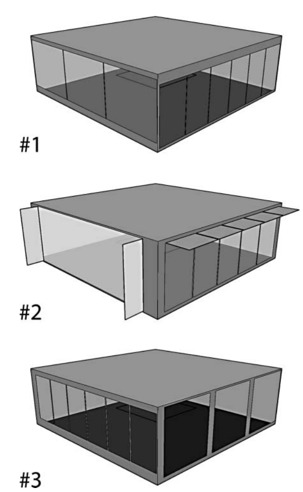


Figure 8. Final solutions for each of three trials for case study with no maximum illuminance.

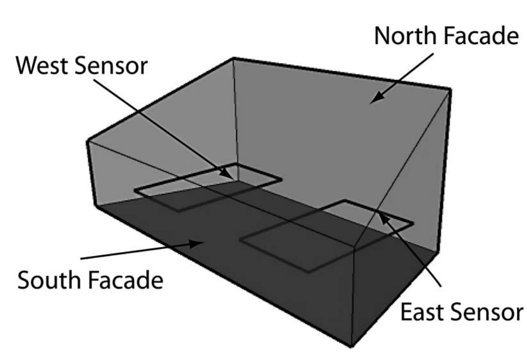


Figure 9. Original massing model for case study with two non-conflicting illuminance goals.

work day, on the author's computer. The fitness in this case study was calculated as the mean of the goal-based illuminance metric for both sensors. Therefore, a value of 100% would indicate that the entire area of both sensors would be within the specified illuminance ranges throughout the whole year. The population average and best fitness for each generation are shown in Figure 10. The average fitness decreases at certain generations (generations 6, 10 and 20) due to the bitwise convergence and subsequent re-initialization of the population. After 25 generations, the best solution was found to have an average fitness of 90.2% (individual fitnesses for the two sensors were 96.7% and 83.7%). The final solution facades both have windows concentrated towards the west size of the space as expected based on the specified goals (Figure 11). The final solution found is less than expected for an "optimal" design, which indicates that the two performance goals considered may not have been completely non-conflicting, i.e. a solution with a fitness of 100% may not exist. However, from Figure 10, we also note that the average fitness was increasing and had not yet converged once 25 generations were completed, which means that a better solution may have been found if our stopping criteria had allowed the GA to continue past 25 generations.

4.2. *Case study no. 2: conflicting illuminance goals*

While the previous case study considered two non-conflicting goals, it is possible that a designer might need to consider designs which have conflicting illuminance goals. Therefore, the second case study considered a problem in which two illuminance goals were considered conflicting and which used a multi-objective approach rather than a single-objective algorithm. In this case study, a pseudo-Pareto front of non-dominated solutions representing a range of different designs is obtained instead of a single solution.

For this case study, the massing model shown in Figure 12 was considered. The model has a U-shaped footprint with two sensor planes, one located towards north and one located towards south. The goals for the north sensor are 0 lux (acceptable) and 200 lux (desired) minimum; 500 lux (desired) and 700 lux (acceptable) maximum. The goals for the south sensor are 800 lux (acceptable) and 1000 lux (desired) minimum; no maximum illuminance. The facades of interest in this case study are the north, west and south, and the reason that the goals are considered conflicting is that an additional constraint is added to the problem in which all facades must have a uniform aesthetic. To enforce this constraint, the same binary string was used for all three facades.

The multi-objective micro-GA was run for 50 generations and a pseudo-Pareto front was generated. In total, the simulations required for 50 generations took approximately 9 h, or a full work day, on the author's computer. We assumed that in a design

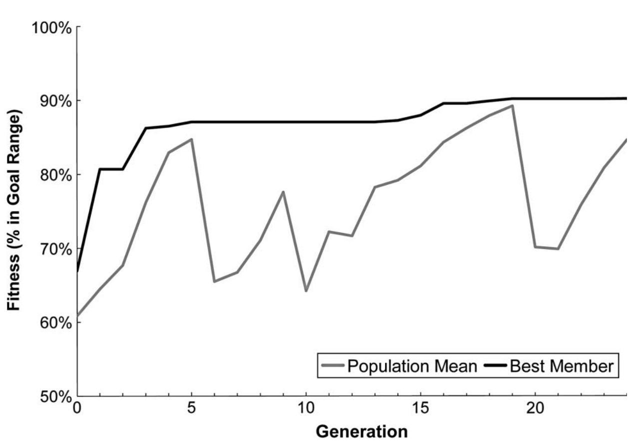


Figure 10. Population best and mean fitness over 25 generations for case study with two non-conflicting illuminance goals.

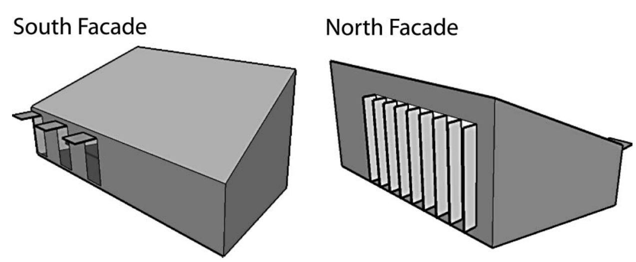


Figure 11. Final solution for case study with two non-conflicting illuminance goals.

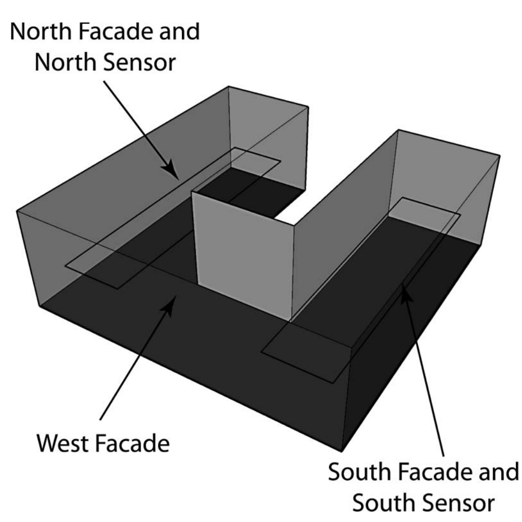


Figure 12. Original massing model for case study with two conflicting illuminance goals.

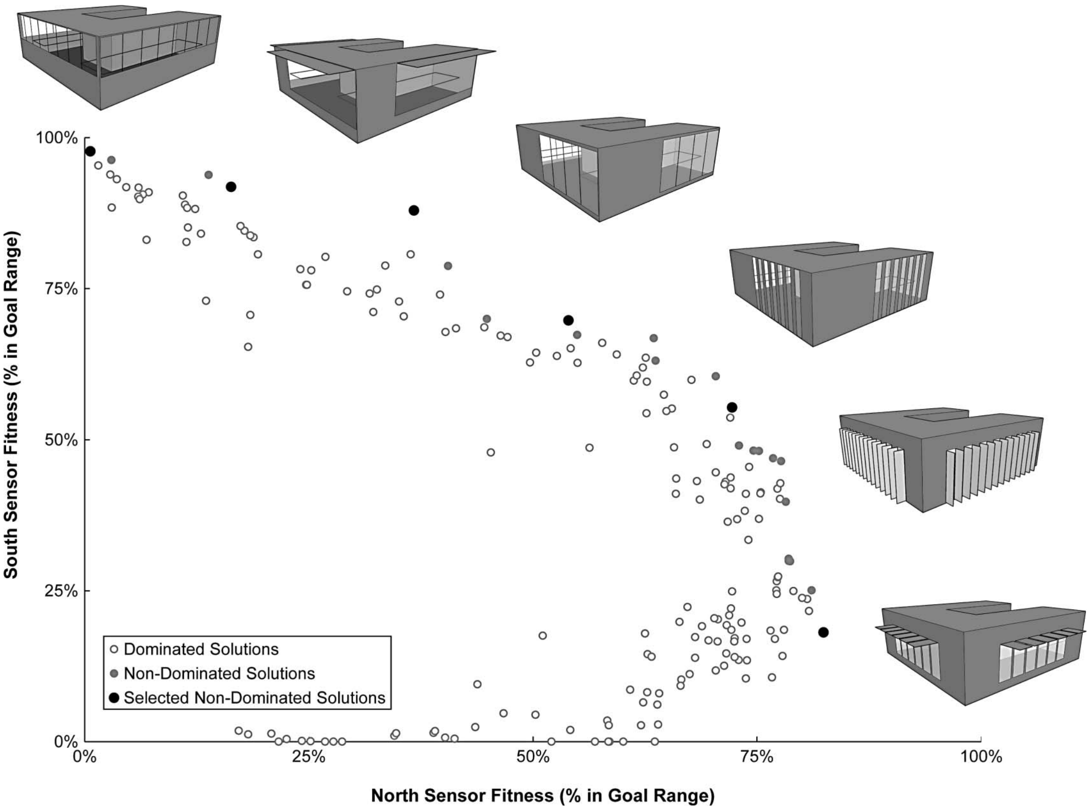


Figure 13. Fitness for all generated solutions (50 generations) for case study with two conflicting illuminance goals, with subset of selected non-dominated solutions.

scenario, a designer would be unlikely to consider running the system for any number of generations that would require more than one full work day to complete.

A subset of six non-dominated solutions are highlighted to indicate the variety of designs generate by the GA-based method (Figure 13). While it is clear that designs with large window areas belong on one end of the pseudo-Pareto front and designs with small window areas belong on the other end, the designs in between represent interesting choices for designers who are trying to compromise between the two illuminance goals. For example, several of the middle designs have windows on the west facade which are shifted towards the south. These designs meet the constraint that all facades must have the same aesthetic while providing more light to the south sensor. Because the goal range for the south sensor is much less restrictive than the goal range for the north sensor, we note that the non-dominated solutions found include those which approach 100% fitness for the south sensor; however, the maximum value found for the north sensor was 82%. While a large window area seems sufficient to meet the south sensor illuminance range goals, those designs which come closest to meeting the north sensor illuminance goals use a combination of smaller window area, shading devices and low-transmissivity glazing to control daylight such that the narrow illuminance goal range on the north sensor is met.

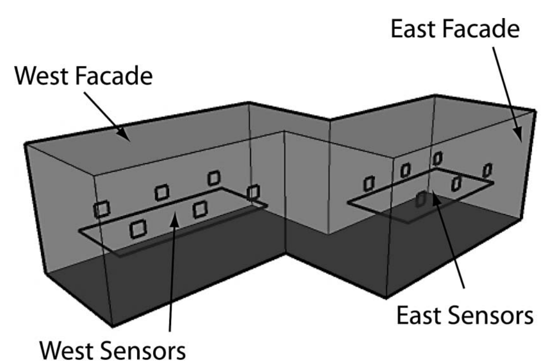


Figure 14. Original massing model for case study with conflicting illuminance and glare goals.

4.3. *Case study no. 3: conflicting illuminance and glare goals*

The multi-objective approach was applied to the massing model shown in Figure 14 in Boston, MA.

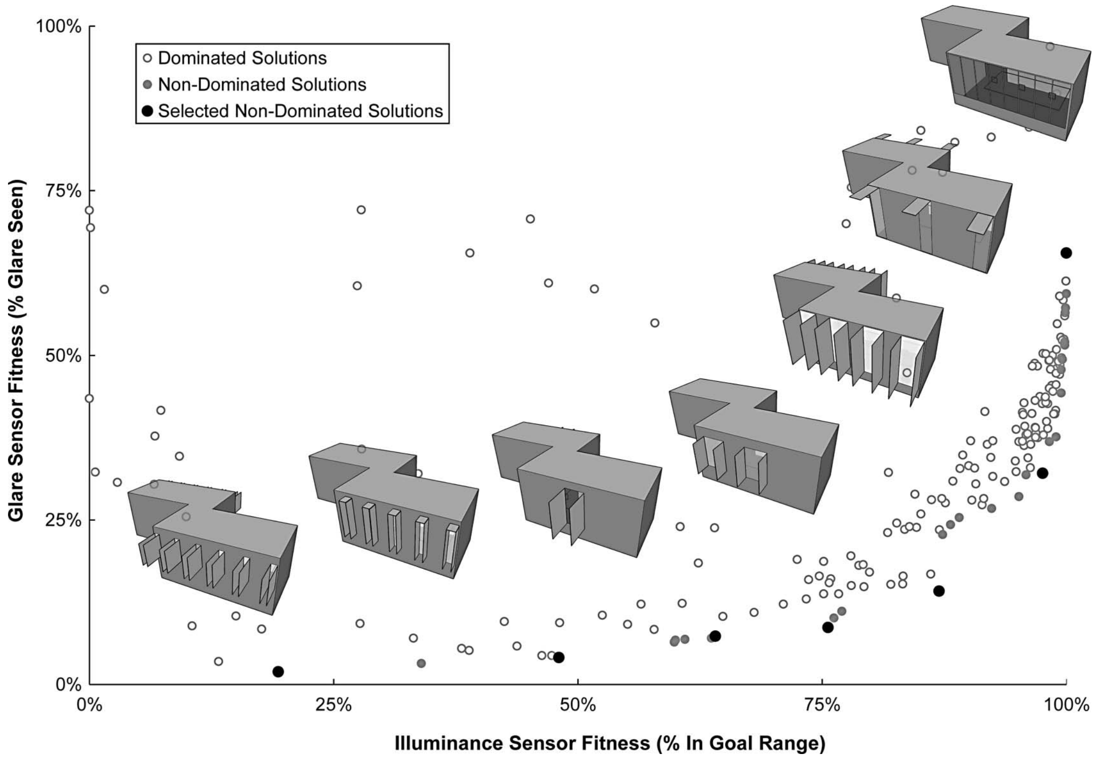


Figure 15. Fitness for all generated solutions (50 generations) for case study with conflicting illuminance and glare goals, with subset of selected non-dominated solutions.

In this model, the two facades of interest are facing east and west. Similarly to the previous case study, an additional constraint is added to this problem, i.e. the two facades of interest must maintain a uniform aesthetic. Two illuminance sensors are included, each with the same illuminance goal ranges (200 lux acceptable low, 400 lux desired low, no maximum). Additionally, glare sensors facing towards the east and west facades are considered. These sensors are indicated in Figure 14. The glare threshold for this problem was set to "zero", i.e. a calculated DGPm value above 0.37 on a given sensor indicates 100% glare in that direction. This threshold value was chosen as it is the strictest glare threshold, and thus would be the most difficult to satisfy.

A pseudo-Pareto front was created after running the micro-GA process for a total of 50 generations, as indicated in Figure 15. It is clear from the pseudo-Pareto front that the two goals are conflicting, although many designs have been found which come close to meeting the illuminance goals while keeping the glare low. A subset of seven non-dominated solutions has been selected to show the variety of solutions found (Figure 15), and we note that many of these non-dominated solutions include vertical fins as shading devices, which is expected due to the east and west orientations of the facades. From this case study, the designer can begin to understand that designs with vertical shading devices combined with smaller window area and low-transmissivity glass have lower potential for glare situations than designs with large window area, high-transmissivity glass, horizontal shading devices or no shading devices.

5. Conclusions

This article presents a GA-based approach which enables performance-based exploration of facade designs. This method combines a micro-GA algorithm with an intuitive set of user inputs, including an original 3d massing model and user-specific performance goals. Such an approach is powerful because it allows an infinite number of possible design scenarios to be considered using the same system. In doing so, it allows users who only have modelling experience to use GAs during the design process. It also provides a way for designers to explore the trade-offs between performance and form by trying a variety of initial massing models and performance goals.

Several case studies were presented which showed the performance of the single and multi-objective micro-GA search processes. The multi-objective case studies in particular demonstrate the range of possible design solutions that a user can obtain using a set of non-dominated solutions. In all single-objective cases, the GA method found one or more solutions that approach the goal conditions. The case studies also demonstrated the variety of massing models and performance goals that can be considered using the proposed goal-oriented approach. Although these case studies represented but a small subset of the wide range of possible design problems that could be considered, it should be clear that the system is a successful GA-based method which is easily customized to specific problems.

The proposed GA-based approach still has several limitations. One of these is the lack of consistency in the final solutions found, since the randomly generated initial design solutions play a large role in determining which subsequent designs are found. This limitation can be solved to some degree by running many generations, but this approach adds additional time to an already time-consuming process. One other limitation is the tendency for the micro-GA to get "stuck" in a solution that is only a local minimum or maximum. This behaviour is due to the implementation of the micro-GA with a very small population size and the limited number of generations that were completed for the presented case studies. However, for the purposes of early stage performance-based design exploration, it is not necessary to find a global optimum; rather, it should be sufficient to present the user with a design or set of designs which the user will then use as an initial design rather than a final one.

Although it is unlikely that a designer would completely accept a solution generated by a GA, the method has much potential in that it can begin to inform the designer about facade conditions which are more likely to result in good performance than other possible designs. As the proposed method allows consideration of both illuminance and glare performance, designers can evaluate daylighting from both energy and visual comfort perspectives. Because the method uses 3d models, there is also the possibility that one could connect it to a thermal energy simulation engine to consider additional objectives as well. Although GA methods should not replace the traditional design process, the proposed method has the potential to automate parts of the design exploration process in a way which may provide surprising results to designers and which may ultimately influence them to consider performance earlier in the design process.

The approach demonstrated in this article is a first step towards integrating an intelligent search method into the design process. Because the framework has already been created, future modifications or additions to the system may be quickly implemented. These additions could include more specific stopping criteria based on population convergence, more facade parameters including advanced fenestration materials and internal shading, and the ability for the user to add constraints to the design parameters. Future work will also focus on the development of a search method which includes daylighting expertise to improve efficiency and provide more educational value to the user.

Note

1. The author's computer uses a 2.66 GHz Intel Core 2 Quad processor and 4 GB of SDRAM at 800 MHz.

References

Caldas, L., 2008. Generation of energy-efficient architecture solutions applying GENE_ARCH: an evolution-based generative design system. *Advanced Engineering Informatics*, 22 (1), 59–70.

Caldas, L.G., and Norford, L.K., 2002. A design optimization tool based on a genetic algorithm. *Automation in Construction*, 11 (2), 173–184.

Carroll, D.L., 1996. Genetic algorithms and optimizing chemical oxygen-iodine lasers. *Developments in Theoretical and Applied Mechanics*, 18, 411–424.

Charron, R., and Athienitis, A.K., 2006. Optimization of the performance of double-facades with integrated photovoltaic panels and motorized blinds. *Solar Energy*, 80 (5), 482–491.

Chutarat, A., 2001. *Experience of light: the use of an inverse method and a genetic algorithm in daylighting design*. Thesis (PhD). Massachusetts Institute of Technology.

Coello Coello, C.A., Lamont, G.B., and Van Veldhuizen, D.A., 2007. *Evolutionary algorithms for solving multi-objective problems*. 2nd ed. New York: Springer.

Coello Coello, C.A., and Pulido, G.P., 1993. A micro-genetic algorithm for multiobjective optimization. *Lecture Notes in Computer Science*, 1993, 126–140.

Cutler, B., *et al*., 2008. Interactive selection of optimal fenestration materials for schematic architectural daylighting design. *Automation in Construction*, 17 (7), 809–823.

Geyer, P., 2009. Component-oriented decomposition for multidisciplinary design optimization in building design. *Advanced Engineering Informatics*, 23, 12–31.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Google, 2010. *Google SketchUp* [online]. Available from: http://sketchup.google.com [Accessed August 2010].

Greenspace Research, 2007. *Demeter – The SketchUp Green Building Extension* [online], Greenspace Research. Available from: http://www.greenspaceresearch.com/demeter_tutorial1.html [Accessed August 2010].

Kleindienst, S., 2010. *Time-varied daylighting performance to enable a goal-driven process*. Thesis (PhD). Massachusetts Institute of Technology.

Kleindienst, S., and Andersen, M., 2009. The adaptation of daylight glare probability to dynamic metrics in a computational setting. *In: Proceedings of LuxEuropa 2009*, Istanbul.

Kleindienst, S., Bodart, M., and Andersen, M., 2008. Graphical representation of climate-based daylight performance to support architectural design. *Leukos*, 5 (1), 39–61.

Krishnakumar, K., 1989. Micro-genetic algorithms for stationary and non-stationary function optimization. *SPIE Proceedings: Intelligent Control and Adaptive Systems*, 1196, 289–296.

Lee, J., *et al.*, 2009. Goal-based daylighting design using an interactive simulation method. *In: Proceedings of Building Simulation 2009*, Glasgow, Scotland, 936–943.

Park, C., *et al.*, 2004. Real-time optimization of a double-skin facade based on lumped modelling and occupant preference. *Building and Environment*, 39 (8), 939–948.

Shea, K., Sedgwick, A., and Antonuntto, G., 2006. Multi-criteria optimization of panelled building envelopes using ant colony optimization. *Lecture Notes in Computer Science*, 4200, 627–636.

Torres, S.L., and Sakamoto, Y., 2007. Facade design optimization for daylight with a simple genetic algorithm. *In: Proceedings of Building Simulation 2007*, Beijing, 1162–1167.

Vartiainen, E., Peippo, K., and Lund, P., 2000. Daylight optimization of multifunctional solar facades. *Solar Energy*, 68 (3), 223–235.

Wang, W., Rivard, H., and Zmeureanu, R., 2005. An object-oriented framework for simulation-based green building design optimization with genetic algorithms. *Advanced Engineering Informatics*, 19, 5–23.

Wienold, J., 2009. Dynamic daylight glare evaluation. *In: Proceedings of Building Simulation*, *2009*, Glasgow, 944–951.

Wienold, J., and Christoffersen, J., 2006. Evaluation methods and development of a new glare prediction model for daylight environments with the use of CCD cameras. *Energy and Buildings*, 38 (7), 734–757.

Wright, J., and Mourshed, M., 2009. Geometric optimization of fenestration. *In: Proceedings of Building Simulation 2009*, Glasgow, 920–927.

Appendix. Building model population logic

This logic applies to the initial massing model that the user creates in Google SketchUp. All material names may be specified within SketchUp. In order for massing model to be correctly identified by the system, the following guidelines must be met:

(1) Any plane that represents a sensor (for either illuminance or glare) must have the word "SENSOR" in its material name.
(2) Any plane that represents an external shading device must have the word "EXTERNAL" in its material name.
(3) Any plane that is manipulated by the GA must have the words "GA_WALL" in its material name.
(4) The normal vectors of all faces should point towards the interior of the space.

Assuming these guidelines are met, the logic used to identify each element is as follows (assume all elements are faces):

(1) If the face is not opaque and not called "SENSOR", it is a window.
(2) If the face is opaque and called "EXTERNAL", it is a shading device.
    (a) If the normal points up or down, it is an overhang.
    (b) Else, it is a fin.
(3) If the face is opaque and not called "EXTERNAL":
    (a) If the normal points up, it is a floor.
    (b) If the normal points down, it is a ceiling.
    (c) Else, it is a wall.
(4) If the face called "SENSOR", it is a sensor plane.

Once the individual building elements have been identified, a second set of logic is used to determine the appropriate relationships between elements. This logic determines the child–parent relationships between walls and windows and between windows and shading devices. The logic for determining these relationships is as follows:

(1) Assigning windows to walls: For each window, cycle through all walls. If both elements have the same orientation, and if the window location lies between the edge boundaries of the wall, assign that window to that wall.
(2) Assigning shading devices to windows: For each shading device, cycle through all windows. If two vertices of the overhang is located 0.05 m [2 inches] or less from two vertices of the window (top two vertices for overhangs, right or left vertices for fins), assign that shading device to that window.

An initial massing model may or may not include windows and shading devices. If the model does include these elements, they will remain the same through the GA process. Only those walls that have been labelled "GA_WALLS" will have generated facades.