Agent-based Geometry Control Optimized by Genetic Algorithm for Daylighting

Yun Kyu Yi[1*], Bo Rin Jung[1] and John Sullivan[1]

[1]Department of Architecture, School of Design, University of Pennsylvania,
210 South 34th street Philadelphia PA, USA 19104

ABSTRACT
This paper proposes a method for daylighting performance-driven building fa ҫade generation. The daylighting condition for the building is considered as a domain and is evaluated by advanced light simulation. Forms are generated by parametric CAD toolsand this form is optimized with genetic algorithm (GA), with the objective to find better indoor daylighting conditions. Using the proposed method, engineers and designers may optimize the building configuration to solve the complex combinatorial problems of designing illuminated spaces. GA optimization requires the computationally expensive process of generating a large number of simulation models automatically. This paper documents the implementation of an agent-based control system which decreases the number of daylight simulations required to reach an optimal configuration. The set of parameters that define the fa ҫade are controlled in a hierarchical system by a relatively small number of control points or "agent points." This study demonstrates the scalability and feasibility, as well as reduces the cost in time and processing power of the method.

KEYWORDS
Genetic Algorithm(GA), Optimization, Daylighting, Radiance, Agent-Based Geometry

1. INTRODUCTION
A genetic algorithm (GA) is a heuristic search algorithm for solving constrained optimization problems with clearly defined parameters and objectives. GAs belong to a larger group of biologically-inspired tools in a branch of computer science called Evolutionary Computation. The concept of using computers to simulate biological evolution, or more accurately selective breeding, was proposed since the existence of computers. In 1948, Alan Turing proposed a GA for artificial neural network training, and expanded this idea [1] and by Fogeland Holland [2] developed the modern evolutionary computation [2]. The attempt to implement evolutionary optimization tools to architectural design was pioneered by Antony Radford and John Gero among others [3].

Architectural application has been demonstrated in several studies [4, 5, 6, 7, 8, and 9]. Genetic algorithms are particularly useful when there is a complex relationship between parameters and results, or between differing parameters. Due to a large number of variables, previous parametric studies using simulation-driven GAs to optimize glazing have achieved good results

[*] ykyi@design.upenn.edu

but required substantial time and processing power [8 and 10].Various methods to address the computational limitations of evolutionary optimization have been proposed, such as strategic placement of individuals in the initial population [11], modifying the search algorithms to converge more quickly [12], and simplifying the environment or reducing the search space [8 and 13].

The purpose of this study was to develop and demonstrate the form finding ability of simulation-driven GAs for fa çade optimization with agent-based controlto optimize the indoor light level with an advanced simulation tool.The method proposed in this study was connected to a hierarchical agent-based control system that effectively reduces the number of generations needed to find an optimal configuration, allowing more variations to be explored by the GA.

The agent based method of control was applied to two case-studies, simple box with one window on the southern fa çade and a multi-use office space with multiple windows in South fa çades. The first case study was to compare the proposed agent-based geometry control method with the conventional geometry control method to show the effectiveness of the proposed method. The second case study applied the proposed method to the complex problem in order to demonstrate its strengths fora real practical project.

2. METHODOLOGY

The overall process is illustrated in Figure 1, for which the initial building form will be modeled with surrounding buildings on a commercial district. This geometrical information of a built environment in NURBS modeling software will be evaluated by an advanced lighting simulation to determine whether the indoor light level is comfortable to residents or not. The next step is to create the alternative window shape by modifying the initial geometry, which will be analyzed by an advanced lighting simulation for the change of illuminance, as compared to the objective function. At this stage, the better performing building form(s) will influence the next generation as in genetic algorithms (GA), alternative forms will be produced until either they converge or the predefined number of generations are achieved.

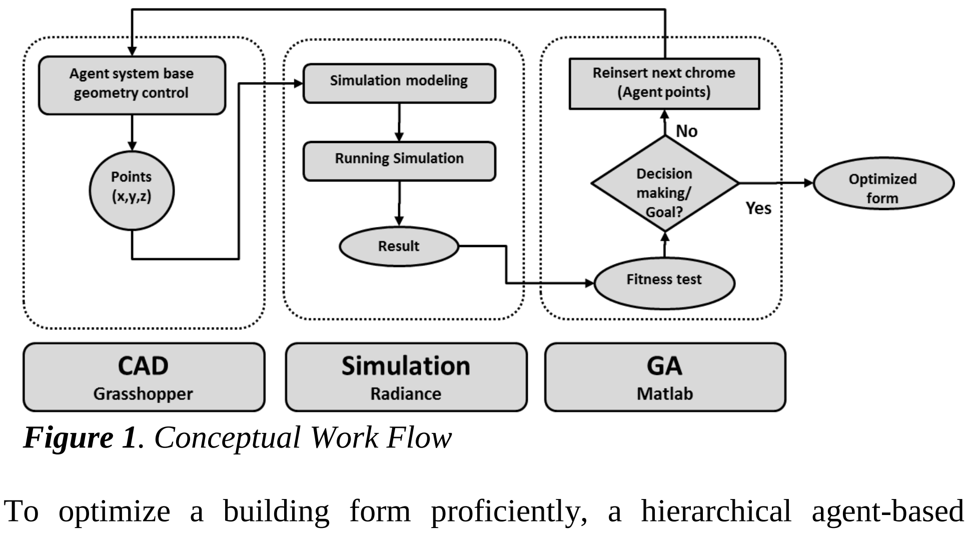


***Figure 1**. Conceptual Work Flow*

To optimize a building form proficiently, a hierarchical agent-based control system was developed building on the author’s previous studies [9]. Design parameters are organized into a hierarchical system of nodes within a search space. Using this system, complex geometric forms may be generated with a limited number of variables.The ability to manipulate forms as objects with hierarchical relations is of great importance to developing a new representation that can be integrated with an optimization model. When introducing the deformed (3-dimension) shape to the optimum method, the number of geometry variables increases and it is difficult to relate geometry variables to optimization, which brings the issue of controlling the variables. To overcome the above challenge a new approach was developed.

The new developed method introduces a point system that controls child points, Figure 1. In this approach an "agent" point controls the position of the child points; when an agent point moves from position a(x,y) to position b(x,y), that movement also changes the positions of its child points. This method allows morphing the building form with a few agent points rather than multiple individual points.GA utilizes a stochastic global search in order to become an optimized method that mimics the natural biological evolutionGA operates on generating potential solutions by applying the principle of survival of the fittest and successively produces better approximations to a solution [14].GAs require automation of all steps of the process. Once the design and simulation tools are linked, single-point crossover GAs are run to control the optimization, giving the designer access to a large set of alternative customizable forms.

3. CASE-STUDIES

To test the strength of the proposed method, an architectural daylighting problem was chosen to be optimized by a lighting simulation-driven genetic algorithm for a case study. As in Figure 1, Grasshopper was used as a tool to create and control the geometry, Radiance was used to simulate light condition of the inside, and MATLAB was used for optimization.

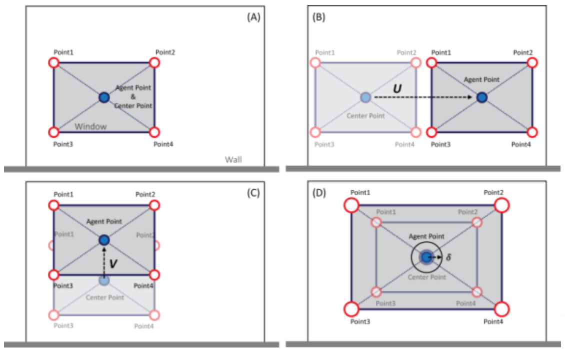
Integration with NURBS software provides ready graphic representation, as well as access to a wide array of tools developed for use with these software suites. Integration with programming suites facilitates easy record keeping for later analysis, GA modification, and code optimization. Improving access to multiple computer languages and design tools is useful since the system is interconnected, and automatic access to a new component by any one existing component provides another tool. This system may be used to govern the CAD and simulation tools for parametric optimization of material properties, geographic location, weather conditions, and geometric configuration from the scale of fa çade elements to the whole building including site modifications.

3.1.Case Study 1: Agent Control

To determine the efficiency and effectiveness of the agent system, two tests were conducted with an agent point that has a hierarchical control and a conventional direct control. Both tests have an identical setting for optimization (i.e. identical mutation, crossover rates, and identical selection pressures) and the same objective function.

For an agent point hierarchical control, a window was controlled by the agent point as shown in Figure 2. The agent point controls the four child points (Point 1, 2, 3, and 4), if the agent point moves horizontally, the magnitude of displacement (U) dynamically alters the geometry and updates the child points, which moves the location of the window.The same method can be applied vertically as portrayed by V, whichshows the window moving vertically, and as the scale factor (δ) changes the window size also changes. With three variables the window can be controlled without controlling each separate four points.

For the case study, both tests have 5 individual (N_5) and for the agent point hierarchical control has three genotypes (V, U, and δ), (L3) and conventional direct control has eight genotypes (X_{p1}, Y_{p1}, X_{p2}, Y_{p2}, X_{p3}, Y_{p3}, X_{p4}, and Y_{p4}), (L8). The solution domain size for the agent point hierarchical control was 5×3 matrix and conventional direct control was 5×8. Both test cases have the same simulation conditions. The test site contains four surrounding buildings with different heights. Figure 3 shows the dimensions of the surrounding buildings. The site is located at latitude of 40.1, longitude -75. The test building size is $13{,}000 \times 11{,}000 \times 10{,}000$mm. For the lighting simulation boundary, the sky was set as sunny on December 21st at noon time. The opaque surfaces of buildings were assigned with the same RGB value and windows with a 0.65 transmittance.



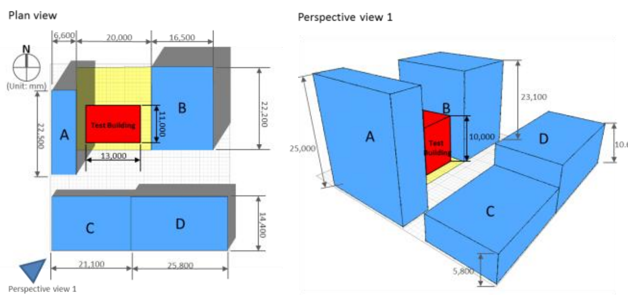


***Figure 2**. Agent point controlFigure 3. Test site*

The objective for both tests was to find the optimum window size and location that allows proper natural light for indoor space while considering the impact from the surrounding buildings. Based on the author's previous paper [9], the object function for case study one can be written as below. Both tests used the same objective function to find the average illuminance light level, less than 1000 Lux, and the average illuminance light level (r_{mesh}) is the average of 576 mesh points (24×24 mesh) located at a height of 1000 mm from the floor.

$$\min F(x) < tF(x)$$

$$F(x) = \frac{\sum_{j=1}^{n} (r_{mesh})_n = (r_{mesh})_1 + (r_{mesh})_2 + \cdots + (r_{mesh})_n}{n}$$

where,

$F(x)$: Object function

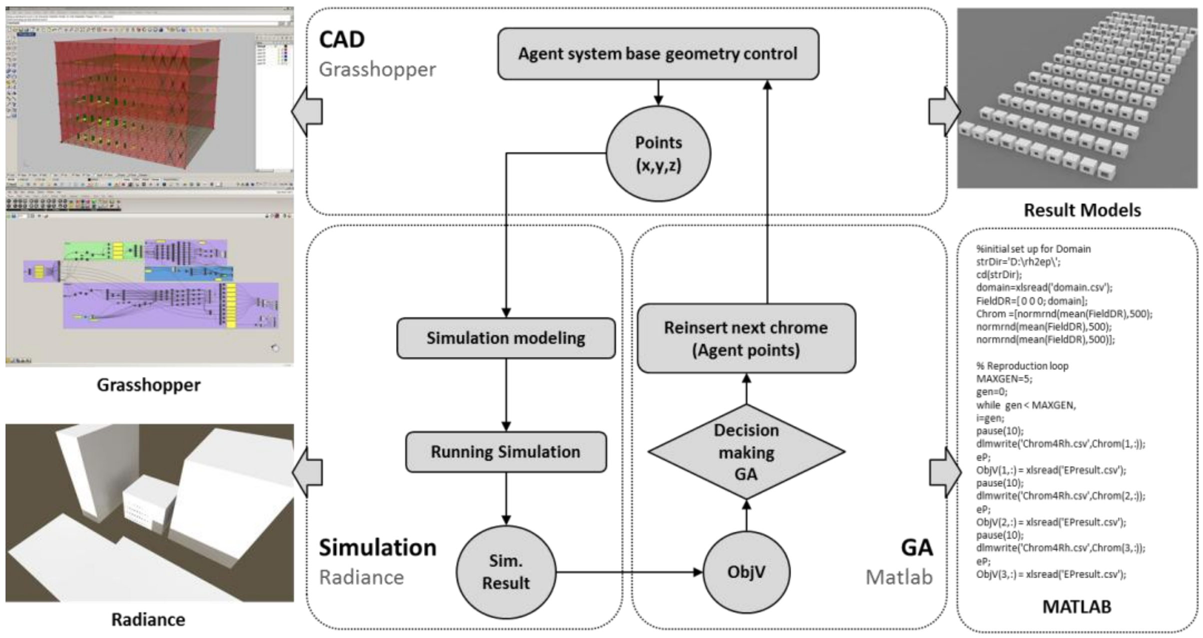
$tF(x)$: target objecetive result value

n: number of mesh points

r_{mesh} : light level

In order to limit the size and location of the window, the following tests were done: for test one with the agent point hierarchical controls, range of U, V, and δ were set to not exceed the boundary of the south fa çade (parent object), and for test two with conventional control range of X_{p1}, Y_{p1}, X_{p2}, Y_{p2}, X_{p3}, Y_{p3}, X_{p4}, and Y_{p4} were set to move only the assigned area.

Figure 4 shows a screenshot of the proposed process for optimization and the selected iteration of optimization. The result shows that test one with agent point hierarchical controls takes 28 iterations to reach the goal of the optimization and for test two with conventional control requires 628 iterations to reach the goal. Including the simulation time in total hours to find the optimal form, test one takes about 2.7 hours and test two takes about 61.9 hours, which is 23 times more than test one (Table 1).



```
%initial set up for Domain
strDir='D:\rh2ep\';
cd(strDir);
domain=xlsread('domain.csv');
FieldDR=[ 0 0 0; domain];
Chrom =[normrnd(mean(FieldDR),500);
normrnd(mean(FieldDR),500);
normrnd(mean(FieldDR),500)];

% Reproduction loop
MAXGEN=5;
gen=0;
while  gen < MAXGEN,
i=gen;
pause(10);
dlmwrite('Chrom4Rh.csv',Chrom(1,:));
eP;
ObjV(1,:) = xlsread('EPresult.csv');
pause(10);
dlmwrite('Chrom4Rh.csv',Chrom(2,:));
eP;
ObjV(2,:) = xlsread('EPresult.csv');
pause(10);
dlmwrite('Chrom4Rh.csv',Chrom(3,:));
eP;
ObjV(3,:) = xlsread('EPresult.csv');
```

Figure 4. *Grasshopper script and visualization of illuminance.*

	Test one	Test two	Difference
iteration	28	628	22.4
time (hrs)	2.65	61.9	23.4

Table 1. *Number of iteration and time to reach optimal goal*

3.2. Case Study 2: with Complex Problem

With the first case study, introducing the control system allows a reduced optimization process that can be utilized in a real professional project. For this reason the second case study was conducted to find how an efficiently proposed method can be applicable to a real project.

The site is located at latitude 36.2 and longitude 140.4, the test building is surrounded by high rise buildings and the surrounding condition is shown in Figure 5. The target building has five stories and it is southwest oriented. The test fa çade area is 25,000 ×15,000mm (width and height) and 75 windows (15windows/floor). For the simulation boundary condition, the sky was set as sunny on December 21st at noon time. The opaque surfaces of buildings are assigned with the same RGB value, and window surrounding buildings are set with high reflective and test building windows with 0.65 transmittance.

For an agent point control, 75 windows were controlled by one agent point. This can be set differently; each floor can have its own agent point to control the assigned floor windows. The merit for introducing the agent point based control is allowing flexibility to define control logics. As shown in Figure 6, the agent point's movements alter the child points and subsequently change the shape of the windows. For case study two, the agent point contains four coordinates (horizontal movement (U), vertical movement (V), window scale factor (δ), and density factor (γ)). 75 windows (300 child Point) correspond, if the agent point moves horizontally the magnitude of displacement (U) dynamically alters the location of the windows, in the same way as V moves the window vertically, as scale factor (δ) changes the window size, and as density changes the windows move closer to the density direction (γ).

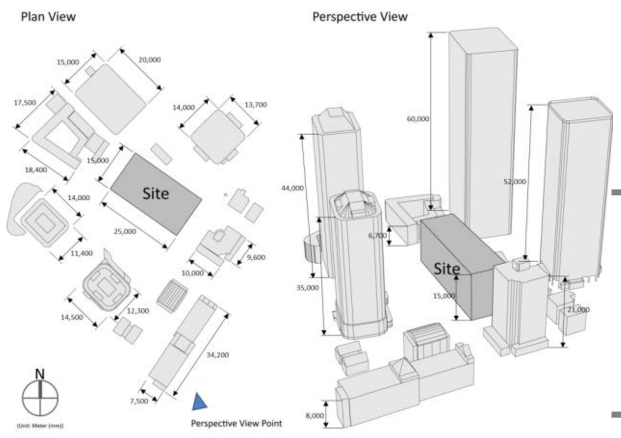


***Figure 5**. Case study 2 site*

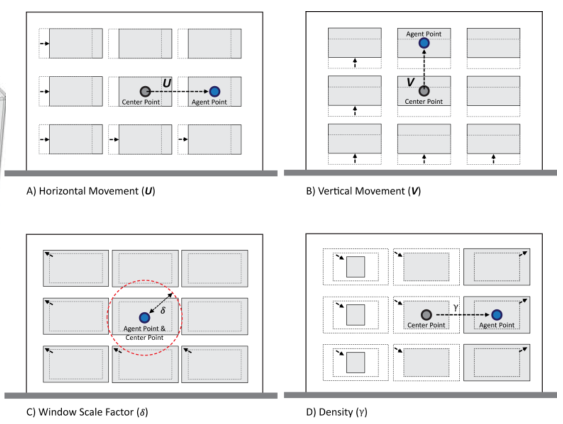

A) Horizontal Movement (U)
B) Vertical Movement (V)
C) Window Scale Factor (δ)
D) Density (γ)
Center Point
Agent Point
Agent Point & Center Point
U
V
δ
γ

***Figure 6**. Agent point control*

The solution domain for case study two has 5 individual (N_5) and each individual has four genotypes (V, U, δ, and γ), (L_4) which makes the solution domain size of 5×4 matrix. If windows were controlled as a conventional direct method, each individual would have 600 (75 windows ×4 points ×2 (x and y) coordinates) genotypes.

$$F(x) = \left| 1 - \left(\frac{\sum_{j=1}^{n} (R_{abs})_n}{n} \right) \right| = \left| 1 - \left(\frac{(R_{abs})_1 + (R_{abs})_2 + \cdots + (R_{abs})_n}{n} \right) \right|$$

when $\min F(x)$

$F(x) < tF(x)$

where,

$tF(x)$: target object value

$R_{abs} = \left| 1 - \frac{R_g}{R_T} \right|$

n : number of floor

where,

R_{abs} : difference from target light level

R_T : target light level of floor

$$R_g = \frac{\sum_{j=1}^{n} (r_{mesh})_n = (r_{mesh})_1 + (r_{mesh})_2 + \cdots + (r_{mesh})_n}{n}$$

where,

R_g : average light level

n : number of mesh points

r_{mesh} : mesh points light level

To make the case study a practical condition, the objective function for case study two introduced the Multi Objective Function. Each floor of the target building has a different target illuminance level as the function of the space needs a different light level (i.e. office space, corridor, living area, lobby, etc need different light levels). Following is the object function for this case study based on our previous paper [9]. R_g is the average light level for the floor. Each floor light level R_g is subdivided by the target luminance of each floor and subtracted from 1 to make it a ratio called R_{abs}. Each floors' R_{abs}were added and subdivided by floor number to get the average ratio. The object value is to find the average illuminance light level that is less than 10% from the target requirement.

Each floor has 528 mesh points (22×24 meshes) at a height of 1000 mm from the floor. The average light level of mesh points of each floor should be closer to the following target illuminance (Table 2).

The limitations for the size and location of the window were set as follows: range of U, V, δ, and γ were set to not exceed the outline of the target façade (parent object).

Floor number	Target illuminance (lux)
1st floor	2500
2nd floor	2000
3rd floor	1500
4th floor	1500
5th floor	1500

Table 2. *Target illuminance of each floor.*

The result shows that the agent point hierarchical controls take251 iterations to reach the goal of the optimization. Based on result of case study one (Table 1), the conventional method to find an optimized form might take roughly 5622 iterations (251 iterations $\times$22.4) or 2124.49 hours (90.79hrs$\times$23.4). This shows the significant computer load reduction that allows finding performative shape or form in the early design stage.

4. DISCUSSION

This study demonstrates the increased flexibility of agent-controlled systems. It is difficult to compare convergence time with the creativity of solutions directly, since creativity is a subjective quality. Using agent points to control geometry in parametric optimization presents a possible solution that allows minimal computer power and time as compared to the direct control of parameters.

In large scale projects, agent-based control can allow for optimization when time and processing power are primary factors. In any project an agent-based GA may provide more flexibility within a fixed timetable than direct variable control, by allowing more optimal solutions to be found at a faster rate. The paper focused on the flexibility of proposed agent point control that lighting simulation result should be more carefully investigate to increase its applicability to real practical practice.

The merit of introducing agent point based control is allowing flexibility to define control logics. Based on the designer's interpretation, form can be morphed as intended and can be able to find the optimized shape based on the performance. Future work will explore this possibility with a multiple objective problem and a more elaborate control system.

BIBLIOGRAPHY

[1] Turing, Alan M., Intelligent Machinery, In "Collected works of A. M. Turing, Mechanical Intelligence," Elsevier Science Publishers, 1992.

[2] Goldberg David E and Holland, John H., "Genetic Algorithms and Machine Learning," Machine learning 3(2-3), pp95-99 .1988.

[3] Antony D. Radford, John S. Gero., "Design by Optimization in Architecture & Construction," New York: Van Nostrand Reinhold Company, 1988.

[4] Caldas, L. G., and L. K. Norford., "Genetic Algorithms for Optimization of Building Envelopes and the Design and Control of HVAC Systems," Journal of Solar Energy Engineering, Aug. (125) pp 343-351, 2003.

[5] Wright, Jonathan A., Heather A. Loosemore, and FarmaniRaziyeh, "Optimization of building thermal design and control by multi-criterion genetic algorithm," Energy and Buildings 34(9) 2002, 959-972.

[6] Znouda, Essia, Nadia Ghrab-Morcos, and AtidelHadj-Al, "Optimization of Mediterranean building design using genetic algorithms," Energy and Buildings 39(2) 2007, 148-153.

[7] Wang, Weimin A., Rivard A. Hugues, and Radu B Zmeureanu, "Floor shape optimization for green building design", Advanced Engineering Informatics, 20(2006) 363-378.

[8] Torres, Santiago L, and Yuzo Sakamoto, "Façade Design Optimization for Daylight with a Simple Genetic Algorithm," Building Simulation 2007, 10th international IBPSA conference, 1162-1167

[9] Yi, Yun Kyu, and Ali Malkawi, "Optimizing building form for energy performance based on hierarchical geometry relation," Automation in Construction 18(6) 2009, 825-833.

[10] Manzan, Marco, and Francesco Pinto, "Genetic Optimization of External Shading Devices," Building Simulation 2009, 11th international IBPSA conference July 27-30 2009, pp180-187.

[11] Mohamed Hamdy, AlaHasan, and Kai Siren. "Applying a multi-objective optimization approach for Design of low-emission cost-effective dwellings," Building and Environment, 46(1) 2011: 109-123.

[12] Wetter, M, and Elijah Polak. "Building design optimization using a convergent pattern search algorithm with adaptive precision simulations," Energy and Buildings 37(6) 2005, 603-612.

[13] Reinhart, Christopher, "Tutorial on the Use of Daysim Simulations for Sustainable Design," Ottawa, Ont., K1A 0R6, Canada: National Research Council Canada, 2006.

[14] Chipperfield, Andrew, Peter Fleming, PohlheimHartmut, and Carlos Fonseca, "Genetic Algorithm TOOLBOX For Use with MATLAB User's Guild," University of Sheffield, 1995.