A framework for generating thermal performance envelope

Hazem Talaat\(^1\), and Ayman ElMasry\(^2\)

\(^1\) Associate professor, Ain Shams university.
\(^2\) Faculty of Engineering, Ain Shams University.

Email: Hazem.eldaly@eng.asu.edu.eg

Abstract. Throughout the last decade, environmental topics in architecture start to gain interest to preserve natural resources, and to improve the performance of designs. One of the great problems facing architects, is checking whether their designs compromise with the environmental needs or not. Thermal performance is one of the main issues related to environmental design. This research proposes a framework for thermal design of envelopes, by simulation and optimization of building envelopes based on computer algorithms and environmental simulation programs. Through analytical methodology, main optimization algorithms and thermal design parameters are analyzed to achieve the framework. It explains the essential concepts of computer programming, and how it is integrated into the architectural design process to achieve thermally efficient envelope. Designer is introduced to computer programming from a theoretical point of view, with an explanation of the types of input needed by the algorithms. In order to achieve this, different types of algorithms and their output, how to define the optimization problem and its variables, the different types of 3D virtual modelling and environmental simulation programs and their role in the process, are discussed. The framework can be summarized in the following steps: building envelope analysis in terms of geometry and its categorization, determining the thermal criteria and target performance, modelling and simulation of the building envelope, and finally applying algorithm and optimization of the envelope’s thermal performance.

Keywords: Environmental architecture, Algorithmic architecture, and Optimization techniques.

1. Introduction

Designing appropriate envelope is a vital issue in architectural design, since it is not only related to the design of form, but it is important in improving or worsening the thermal comfort for the building.

Thermal comfort and internal temperatures of a building space are primarily a function of building envelope design, the building element which controls the inward and outward flow of energy and matter, and architects is faced with the dilemma of balancing the aesthetic element of envelopes with its thermal properties and performance — among other things — for as long as the profession has existed.
The methods used to handle this dilemma is numerous and vary by different conditions such as climate, available materials and resources, available building technologies...etc. Theories have been formulated to assist the architect in his endeavor to control the thermal performance of buildings; however, the results are never full-proof and will always vary due to the large number of variables and constraints that are part of the thermal equation.

With the development of computer applications, new realms are introduced to the process of architectural design to assist in the thermal design of building and particularly building envelopes. The latest is the use of algorithms in the automation generation of building envelopes, through the environmental simulation of virtual building models and passing the results to a program. It evaluates the results and searches for the best solutions by adjusting the variables accordingly. However, as the architect approaches this method, he faces technical difficulties which require knowledge of computer programming, mathematical optimization and physics. This research based on a framework, shows a route for architects to merge complicated algorithms with their applications to automatically examine envelopes performance. And based on examination, alternative for envelopes are generated.

Methodology

To design an envelope with a needed thermal performance; the main issues affecting thermal performance are analyzed and discussed. From the analysis, a framework based on algorithms is achieved.

2. Introducing Algorithms

Algorithms are descriptions of steps to accomplish a specific task, to allow the abstraction of a problem into parameters, or variables, and procedures. They are the detailed instructions given to the computer. The user input values of parameters, and the computer then calculates the outputs according to the given procedure. The problem at hand, is how to use these algorithms to create a framework that would analyze the building form and feedback the optimum solutions to the user (i.e. the architect) [1].

2.1 Building Geometry as Variables

The first step towards achieving this goal is translating the form to inputs, to be translated by the algorithm (the algorithm being basically computer code). The building envelope must be analyzed and broken down into variables and values. It is well known that geometry is built up from points to form lines and planes and finally to build solids. It is preferable to simplify the translation of form as much as possible.

For example as shown in figure1:

The minimum requirement to create a procedure for creating a point in space; can be defined for the algorithm as follows:

- Procedure Handle 1: Point_A
- Parameters/Variables: Cartesian Coordinates (x,y,z)
- Values: (0,0,0) Then, the point procedure would be embedded — or nested — within a line procedure twice with different values for each procedure:
  - Procedure Handle: Line_A
  - Parameters/Variables:(Point_A , Point_B)
Values: [(x,y,z),(u,v,w)] The process is repeated in the same manner until a solid geometry is formed.

3. Algorithmic architecture

It is well known that there are hundreds of predefined algorithms that are applied in many fields. But mainly in architecture, it can be divided into two groups according to their applications in architectural design: Generative and Optimization.

3.1 Generative Algorithms

Architectural generative design is a process which uses rule-based algorithms, governed by repetitive patterns and shape transformations; these algorithms can be fully automated, or user controlled in a step-by-step manner. What differentiates generative algorithms from each other, are the form generating systems which drive the algorithm. It is usually adopted from naturally occurring behavior of living organisms [2].

Prominent generative algorithms used for architectural form generation purposes include:

- Cellular Automata.
- Voronoi Diagrams.
- Fractals.
- Shape Grammars.
- L-System.

Out of the above, two examples are explained briefly here. Cellular Automata is a computational method which can simulate complex growth systems using simple rules, the initial configuration would include at least one cell [7].
Figure 2. An example of cellular automata showing (a) the sequence of generations and (b) the governing rule.

A Voronoi Diagram is a way of decomposing a space into regions, which is a process initiated with a set of points in space called generating points; then, lines are drawn from an equal distance from each two points, creating polygons or cells (Figure 3) [3].

Figure 3. A Voronoi Diagram

Figure 4. National Swimming Center, Beijing, China. By PTW architects
3.2 Optimization Algorithms

The other type of algorithms are called optimization algorithms; which would take a certain aspect of the building performance — thermal performance of the envelope in this particular case — and searches for solutions that would enhance the performance, ultimately finding the best solution. The underlying process of Performative Algorithms is called Mathematical Optimization [3].

Mathematical optimization mathematics and programming is called a search method, which searches for and locates the optimum solution. The optimum solution can be the maximum value of a function, or the minimum value of a function; therefore, all mathematical optimization operations are either called minimization or maximization problems [4].

The general mathematical form of unconstrained optimization is:

$$\min f(x), \ x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R} \quad (1)$$

The continuous components of $x$ are called the design variables, and $f(x)$ is the objective function. The optimum vector that solves the equation is denoted by $x^*$ with a corresponding optimum function value of $f(x^*)$ (Figure 5).

The full process of mathematical optimization is to take a real-world problem (e.g. thermal performance of a building envelope), construct it into a mathematical model of fixed parameters and variables, then process it through the optimization algorithm, leading to the practical implication of the optimum solution (Figures 4 and 5).

![Figure 5. A single variable function with the optimum at $x^*$](image)

**Figure 5.** A single variable function with the optimum at $x^*$ [5]

![Figure 6. The Mathematical Optimization Modelling Process](image)

**Figure 6.** The Mathematical Optimization Modelling Process
Mathematical optimization algorithms are numerous and differ greatly in solution space searching methods. The predominantly common algorithms in architectural applications are:

- Genetic Algorithm
- Simulated Annealing
- Tabu Search

It should be noted that although the three are very different in terms of origin and search method, they have one thing in common which is being under the Metaheuristic class of mathematical optimization algorithms.

### 3.2.1 Genetic Algorithm

Genetic Algorithm Genetic Algorithms are evolutionary algorithms which simulate biological evolution and draws most of its terminology from biological evolutionary science [1].

The process of optimization using GA can be described as follows: “Under GA terminology, a solution to a problem is an individual, and the group of solutions existents at each stage is a population. Each time a new population of individuals is created it is called a generation. In binary GA’s[…]each individual is represented by a binary string called chromosome, which encodes all parameters of interest corresponding to that individual. A chromosome of formed of alleles; the binary coding bits. The fitness of any particular individual corresponds to the value of the objective function at that point” [5].

An initial population is randomly generated, with each of the individuals coded with different genetic design variables (using the notion of chromosomes), then a number of stochastic operators start to manipulate the initial population, including

1. Crossovers; the swapping of parts of chromosome values between different individuals.
2. Mutation; introducing new values randomly to some individuals to create diversity.

Finally, the elitist model is created from the best set of solutions in this generation and is used as the initial population of the next generation. The same loop keeps repeating until the termination criteria of the GA are met and the final set of solutions are recorded (see Fig. 7) [5].

![Flowchart showing stages of GA](image-url)
3.2.2 Simulated Annealing
Simulated Annealing is a mathematical optimization method. It draws its terminology from the metallurgical process of annealing which is the heating and controlled cooling of metals, to reach its ground state with the lowest internal energy. This search method is also Metaheuristic, with emphasis on finding the global maximum or minimum. The process of Simulated Annealing is described by the following algorithm (in pseudocode):

1: Get an initial state with energy $x$
2: Make the initial state the current state
3: Select an initial high temperature $T$
4: while System is not yet frozen do
5: while System is not yet in thermal equilibrium do
6: pick a random nearby state with energy $x_p$
7: let $\Delta x = x_p - x$
8: if $\Delta x \leq 0$ then
9: the new state becomes the current state
10: else
11: reject state
12: end if
13: end while
14: Reduce the temperature $T$ by $\Delta T$
15: end while
16: Output the current state

3.2.3 Tabu Search
Tabu Search depends on what are called adaptive memory and responsive exploration. The basic principle is that it heavily uses different memory-based techniques to;
1. avoid revisiting previously tested solutions,
2. to memorize elements that are common among good solutions,
3. the impact of choices made till a certain point on the progress of the search to find the optimum solution, which is an added level of self-learning.
This makes Tabu Search a very intelligent search method, which is very efficient at finding the optimum solutions in large solution spaces, or to complex problems [6].

4. Thermal performance variables
Having explained the mathematical and basic programming concepts behind optimization algorithms, the next step is to define the function in question; the objective function. In this particular case the objective function is the thermal performance of building envelopes. The exchange of heat between the building and external environment is expressed in the following equation:

\[
Qi \pm Qc \pm Qs \pm Qv - Qe = \Delta S
\]  

$Qi =$ Internal heat gain  
$Qe =$ Conduction heat gain or loss  
$Qs =$ Solar heat gain  
$Qv =$ Ventilation heat gain or loss  
$Qe =$ Evaporation heat loss  
$\Delta S =$ Difference in stored heat
Given that solar radiation is the most significant variable in the above equation, it would be beneficial — for the sake of simplicity — to focus on the aspect of passive solar design when considering the thermal performance of building envelopes.

### 4.1 Passive Solar Control Design Variable

The previously mentioned function variables in optimization problem of building envelope thermal performance through passive solar control, can be any of the architectural elements shown in following table:

| **Shape** | Surface-to-Volume Ratio, Orientation |
|-----------|-------------------------------------|
| **Fabric** | Shading, Surface Material Properties |
| **Fenestration** | Size, Position and Orientation, Glazing Material, Closing Mechanism, External Shading |

**Figure 8.** Variables of building elements to be optimized

Now, the thermal variables that help us to control the thermal performance are defined. The theoretical method of modelling them into a mathematical function is illustrated; and a practical way of measuring and optimizing the thermal performance of the building envelope is discussed in the next step.

### 4.2 Building Modeling and Simulation

There are several methods for modelling and simulation (energy simulation) of building elements. The one discussed herein, and the most practical, is the use of Environmental/Energy Modelling and Simulating computer programs. There are two main advantage of these programs:

These software help the user to overcome calculations, by transferring building elements into thermal equations. These equations are preprogrammed in the simulation application and measures the impact of different variables with different values on the overall thermal performance. The variable values can, later on be manipulated by the optimization algorithm to search for the optimum values through repeated simulation of different values. Thus, reaching the optimum solution for each variable. Also, can perform overall optimum performance in multi-variable cases. Prominent examples of these programs include ecotect, doe, energyplus...etc, among others.

### 5 Framework for generating thermal performance envelope

For the architect to successfully optimize a building envelope’s thermal performance, a clear methodology with correct sequence of action, integrating the above presented information, should be followed. The process can be summarized in the following four stages:

#### 5.1 Stage 1: Analysis of Envelope Form

1. Develop an initial conceptual design
- Analyze the context; building site and conditions.
- Develop the initial building form and type.
- Finalize the conceptual envelope design to include the basic architectural features.
- Assign the developed building form to a predefined category of envelopes.

2. Classify the developed envelope under a general category of building envelopes such as: Tent Structure, Cylindrical Tower, Cuboid Form...etc. This will have implications in the determination of the thermal control mechanisms.

![Envelope Matric Framework]

**Figure 9. Envelope Matrix Framework.**

Assign the developed building form to a predefined category of envelopes. Put the developed envelope under a general category of building envelopes such as: Tent Structure, Cylindrical Tower, Cuboid Form...etc. This will have consequences in the determination of thermal performance.

5.2 Stage 2: Creating Virtual Modelling
Creating the 3D Parametric Model. This model is variable, and later on could be managed with the algorithms in later stages. A model can be generated parametrically through one of the following methods;
- 1. Parametric modelling environments such as Catia.
- 2. GUI enabled simulation programs such as Ecotect. (but the model should be with an embedded script to be easily algorithmically modified).
- 3. 3d modelling software with scripts, or graphical user scripting plugin such as grasshopper.

5.3 Stage 3: Thermal Criteria and Target Performance
The model previously generated is then moved to the thermal simulation program, to test the performance of the generated form.
Assign thermal performance targets at different internal spaces. Based on the requirements of the building and internal space functions, the architect should assign the target temperature levels. These values will act as an input to the optimization algorithm.

Determine the thermal control mechanism. Depending on the category of building envelope, the most effective control mechanism should be chosen for optimization.

### 5.4 Stage 4: Programming and optimization

1. **Determine the programming method**
   - There are two possibilities in this case: The simulation program supports scripting; in this case the algorithm can be programmed directly into the simulation program.
   - The simulation program does not support scripting; the algorithm has to run in other software to show the final results. Then these results are moved back into the simulation program to test them back on the model.

2. **Choose the optimization algorithm from one of the following types:**
   - Genetic Algorithm: Creates a set of final solutions, giving the architect the advantage of choice.
   - Simulated Annealing: Focuses on finding the global optimum.
   - Tabu Search: Can handle very large solution spaces or complex problems.

3. **Initiate optimization and simulation**
   - The optimization algorithm is now programmed, and the virtual model parameters are optimized through the algorithm and each individual solution assessed by simulation.

4. **Termination conditions met**
   - At this point, the target values of the design intent have been met and the optimization algorithm can terminate and present the final results.

---

**Figure 10.** The day-lighting in LUX

---
6. Conclusion
The achieved framework can be summarized in four steps; the first one is designing the main form or envelope for the building. The second step is parametrizing form to be managed by an algorithm. The third one is setting the achieved thermal values from the thermal simulation. The final step is to input the form parametric variables in an optimization algorithm, to achieve the optimum or needed performance.

References
[1] Eleftheria F 2008 Integrated design: A generative multi-performative design approach (Cambridge: Master’s thesis, Massachusetts Institute of Technology, Department of Architecture)
[2] Arida S 2004 Contextualizing generative design (Cambridge: Master's thesis, Massachusetts Institute of Technology, Department of Architecture)
[3] Eva F 2008 The voronoi diagram in structural optimization (London: Master’s thesis, Bartlett School of Graduate Studies, University College of London)
[4] Zubin K 2012 Generative Algorithms (using Grasshopper) (Electronic Material: http://www.morphogenesis.org/).
[5] Kostas T 2006 Algorithmic Design (London: Routledge)
[6] Fred G 1997 Tabu Search (Dordrecht: Kluwer Academic Publishers)
[7] Robert J 2002 Experiments in architectural form generation using cellular automata (Warsaw: eCAADe Conference) p 552