Form Generator as an Alternative Solution for Architectural Layout

Ruşen Eroğlu
Architectural Design Computing Graduate Program, Department of Informatics,
Istanbul Technical University, Turkey
www.itu.edu.tr
e-mail: eroglur@itu.edu.tr

Muzaffer Karslı
Architectural Design Computing Graduate Program, Department of Informatics,
Istanbul Technical University, Turkey

Abstract

This paper introduces a tool called "Form Generator" and its implementation in the context of generative architecture. Form Generator is a software tool that generates spaces for form finding of architectural purposes. The study is based on the concept of autopoiesis which means self-production. Architectural design explores a solution to the initial stage of form finding and form placement. In addition, form finding has crucial importance both in professional and academic practices of architecture, which is extensively carried. This study aims to offer alternative approach exploring generative systems. The used generative systems are genetic algorithms and shape grammars, associated with autopoiesis. As final product, we develop Form Generator tool that assist form finding with rule-based system.

1. Concept of Form Generator

Autopoiesis, the word meaning (auto-self, poiesis-production) is defined as a self-generating system. Maturana and Varela first put forward this concept via cognition perspective; human beings explained that he existed not by human understanding of the world but by creation. Maturana and Varela argued that with this concept, human reproduces his environment with his self-perspective and creates his world by briefly producing self-perception [1].

1.1 Poiesis; beyond praxis

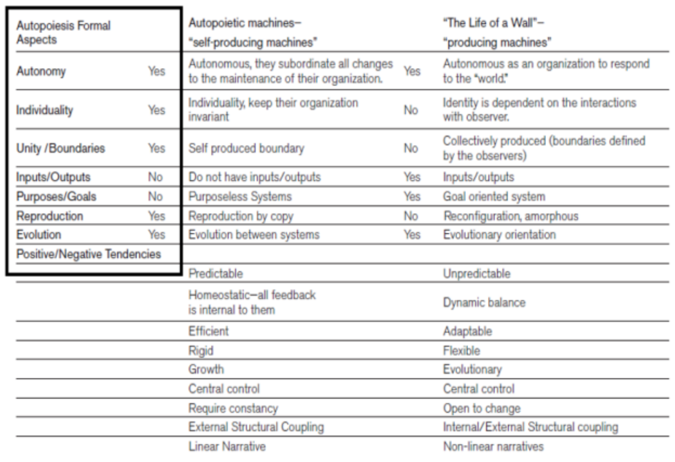
To understand the concept of autopoiesis, we need examine the definition of poiesis. Poiesis is defined as 'making' through the concept of praxis in architectural practice. While praxis is defined as doing an action, poiesis is beyond that and defined as making by act of production. Poiesis can be defined as simple creation in terms of architecture. This includes all kinds of production and growth environment.

Explosion of a flower in the highest sense is exemplified as poiesis [2]. Aristotle emphasis the form-matter relationship in his definition; the poiesis is the material to be included the the form transformed into reality [3]. Maturana and Varela explain that autopoiesis is a cognitive system and describes the cognitive system as living systems, and describe autopoietic systems as an existing structure from the units of cognitive operations. These units provide continuity with interaction and reproduction in a self-directed circular life organization [4].

1.2 Machines and living machines

The machine is a set of combined structures to transform any kind of energy into another energy, to perform specific task, or to create an effect. It contains many systems as well as visible structures. However, although living systems consist of many structures, the point or regional block structures are difficult to see. Another difference is that machines are produced by people through thoughtful, spoken criteria; living machines do not contain constraints due to their existence. [5].

Diniz and Turner compared poietic machines and autopoietic machines (Table 1) when evaluating their 'living wall' study based on the concept of autopoiesis. As Table 1 exhibits, there are many differences in autopoietic machines such as specificity, modularity, and functioning [6].



Autopoiesis Formal Aspects		Autopoietic machines– "self-producing machines"		"The Life of a Wall"– "producing machines"
Autonomy	Yes	Autonomous, they subordinate all changes to the maintenance of their organization.	Yes	Autonomous as an organization to respond to the "world."
Individuality	Yes	Individuality, keep their organization invariant	No	Identity is dependent on the interactions with observer.
Unity /Boundaries	Yes	Self produced boundary	No	Collectively produced (boundaries defined by the observers)
Inputs/Outputs	No	Do not have inputs/outputs	Yes	Inputs/outputs
Purposes/Goals	No	Purposeless Systems	Yes	Goal oriented system
Reproduction	Yes	Reproduction by copy	No	Reconfiguration, amorphous
Evolution	Yes	Evolution between systems	Yes	Evolutionary orientation
Positive/Negative Tendencies				
		Predictable		Unpredictable
		Homeostatic–all feedback is internal to them		Dynamic balance
		Efficient		Adaptable
		Rigid		Flexible
		Growth		Evolutionary
		Central control		Central control
		Require constancy		Open to change
		External Structural Coupling		Internal/External Structural coupling
		Linear Narrative		Non-linear narratives

Table 1. *Comparison of poietic machines and autopoietic machines [6].*

The relations between machines and living systems are also considered in advanced architectural approaches. Recently, the theory of evolution and morphogenesis have provided a long source of inspiration. The simulation of different parametric scenario has adopted widely in computational design instead of genetic code. Hensel, Menges and Weinstock began to investigate this to reveal the differences between self-organization principles, tectonic and thermodynamic systems, even eco-systems and nature and machine production, and the "emerging" properties in life and calculation [7].

2. Methods

2.1 Genetic Algorithms

The genetic algorithm was inspired by Darwinian natural selection in his theory of evolution. Optimization is used in many different disciplines such as machine learning [8]. Genetic Algorithm has been developed for problem solving and optimization where the criteria for solving the problem can be clearly stated.

There are three basic steps in the process of Genetic Algorithm (Figure 1);

Selection of components by criteria,

Crossover to produce new components,

Random mutation of new components.

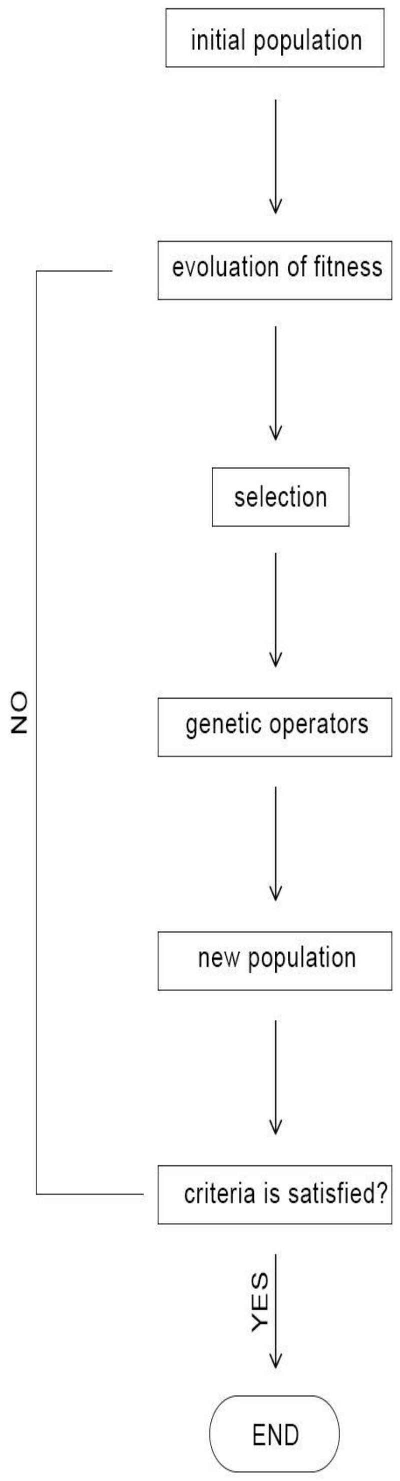


Figure 1. *Simple Genetic Algorithm flow chart.*

Genetic algorithm has been widely used for problem solving in architecture since it is a highly evolutionary and adaptable search procedure that offers various possibilities for architectural evolution and optimization. The design process is described as an undefined problem. Even if the design is coded by designers, this coding will not be sufficient to guarantee an effective and successful design result due to its undefined structure. Genetic Algorithm as an optimization tool can produce a wide variety of possible solutions through crossover and mutation processes. Genetic Algorithm is a self-developed computational method and can find universal solutions [10].

In Jones's GA (Figure 2), he selected lighting, heating and functional criteria [11].

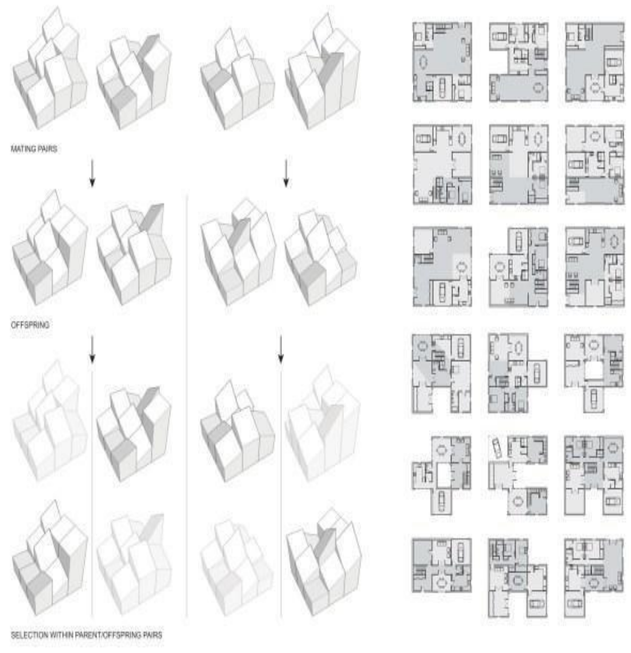


Figure 2. *An example of Nathaniel Louis Jones using Genetic Algorithms [11].*

2.2 Shape Grammars

As an another generative system, Shape Grammar formalism was introduced by George Stiny and James Gips in the early

70s. The linguistic metaphor that permeates form grammar is the basis of Noam Chomsky's work on productive and transformational grammars in linguistics. Their function is to determine design classes with an algorithmic understanding of the processes that make them up.

Shape Grammar consists of the vocabulary of shapes (labelled or unlabelled), a set of shape rules, and a first shape. Rules are presented as transforming shapes or shapes into a new shape or collection of a collection. The first number rules applied with attribute produce designs that are said to belong to a language [12].

3. Form Generator

3.1. Project motivation and purpose

During the development phase of the project, architectural and design problems were considered through the concept of autopoiesis, which was the starting point. In this context, the main objective was to evaluate a design pronoun and to try to solve this problem as a continuous and viable system. One of the most important and fundamental problems among architectural problems is to define the form and mass design. We aim to develop Form Generator system with intend to offer a solution to form definition problem in architectural design process.

3.2. Base shape | square

The initial project process started with a basic form that was formed by thinking about how we would produce a two-dimensional system (Figure 3). At this point, mass production was evaluated in the architectural plan level. The idea of the system, which is produced in two dimensions, can be transformed into a space fiction that meets the architectural requirements has been tried. In this method, the random shape of two points chosen by Genetic Algorithm from the four corner points of the square, which is a fundamental shape, is again provided to realize a random growth (x is defined growth variable by Genetic Algorithm). Due to the undefined and insufficient areas, the desired results could not be obtained. In this method, the random shape is selected by Genetic Algorithm from the four side points of the square which is the base shape and it is ensured that it achieves a random growth again. However, the undefined and inadequate areas, the desired result is not produced.

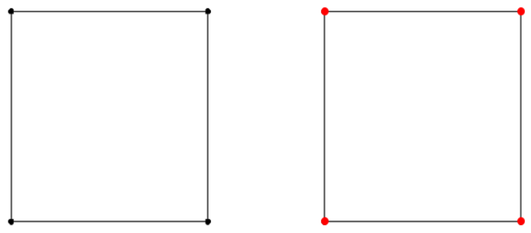


Figure 3. *Base shape, square.*

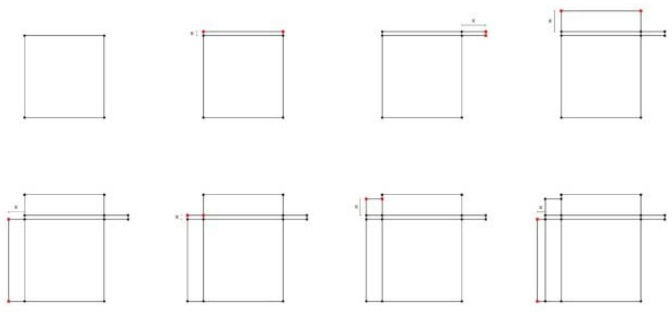


Figure 4. *Base shape, square (growth).*

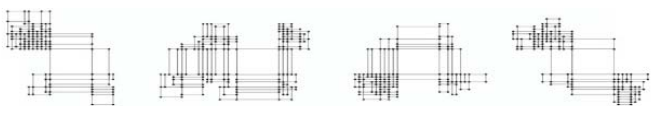


Figure 5. *Alternatives resulting from growth.*

3.3. Base shape | cube

In the second step of the project, the used method was diversified to achieve more effective results and dimensional constraints were introduced to produce meaningful gaps. First, a plane was created to define a certain boundary in the WebGL environment.

Secondly, the cube, which was formed as a unit, was chosen as the first shape to define the meaningful area that would meet the spatial requirements and to invert the growth into 3 dimensions. The process is illustrated in Figure 6.

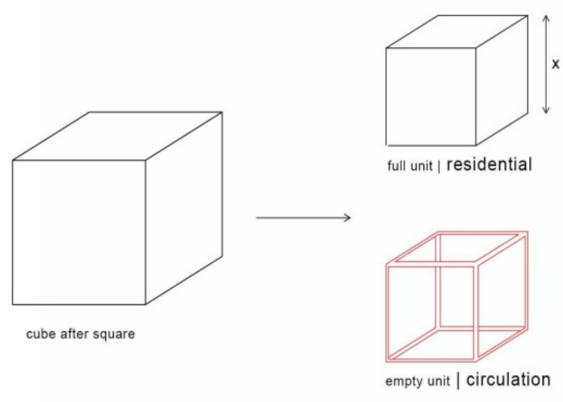


Figure 6. *Base shape, cube.*

At this stage of the project, a growing form of mass design environment has been created with the interaction of the user. In further stages, it will be tried to achieve growth through certain rules. At this stage, we assume that define the growth rules with Shape Grammars and diversification with Genetic Algorithms (Figure 7).

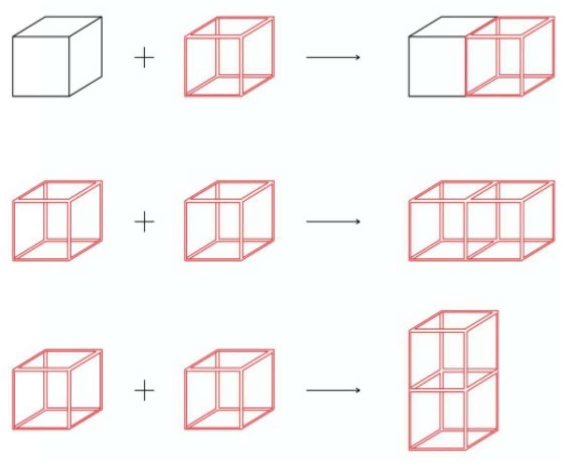


Figure 7. *Neighbourhood rules according to Shape Grammar.*

The application was implemented using HTML, a script-based language. The interaction with the user and the graphic program on the canvas element were performed using the JavaScript language. The user interaction is achieved with an interactive web-based interface (Figure 8).

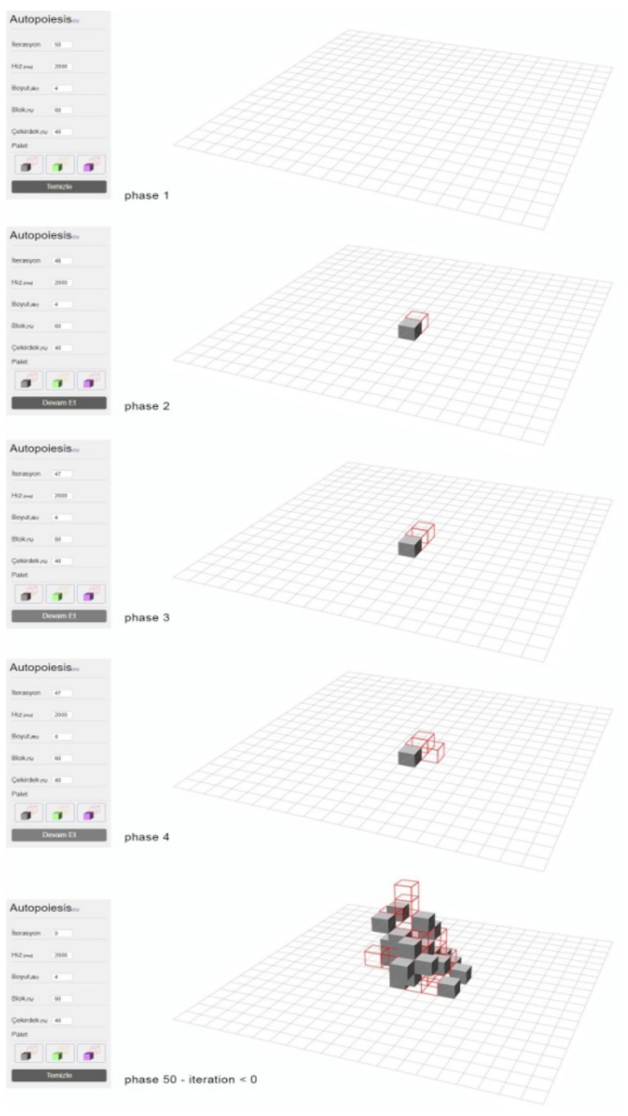


Figure 8. *Iteration phases.*

The interface depends on parameters, which are described below:

• *Iteration:* Refers to the number of repetitions of the algorithm. In other words, the result indicates how many unit elements the product will consist of.

• *Speed:* Specifies the interval at which each iteration repeats.

• *Dimension:* refers to the unit of squares in the plane placed on the three-dimensional plane. Also, the dimensions of the cubes are expressed by the value here.

• *Block:* Specifies the percentage of the block to be produced.

• *Core:* Specifies what percentage of the structure to be produced will be the core.

• *Palette:* The surface colors of the blocks and cores to be produced are presented to the server as three different options.

After the user sets the parameters, the production process starts through clicking on any point on the improvement on the created stage. First, a core element is placed at the point specified by the user. This element is also added to a globally accessible array. Then the direction of the unit element to add is selected randomly. If this direction is relative to the y-axis according to the coordinate system, the new element is determined as the core. Once the type, direction, and axis of the element to be added are determined, the element is placed on the stage with the appropriate coordinates. If the new unit element is the core, it is placed in the global selection array. For following iterations, the random element selection is made by looking at this series. These operations continue until the number of iterations entered by the user. The flowchart of the algorithm is shown in the Figure 9.

4. Results and Discussion

The precise spatial expression of the cube has been effective in the selection of the unit to be reproduced. The growth of the application over a certain volume, such as cubes, has already fed the expression of spatial in terms of architecture. With the size variable added to the interface, that effect has become manageable. The situation has helped to evaluate the outcomes. Thus, the core and settlement unit, which contains some architectural features, can reproduce an architectural meaning by defined rules. The outcomes with this ratio have not been effective enough but promising for the architects.

Several alternatives are developed using this tool (Figure 10). Many design approaches and models have been used within the scope of the project. Along with these, a promising Form Generator application has emerged. Since the tool was designed for the production of an architectural form, it was expected to meet some spatial requirements. These are the necessities of creating functional areas suitable for the user and providing the necessary structural features as in all architectural buildings. Although it does not yet meet the spatial requirements, the tool is promising for architectural use. On the other hand, the biggest deficiency of its insufficient nature is Form Generator is non-contextual reproduction. When an architect designs a building, it not only designs the interior of the building, but also the interaction with the outside. In the application, the environment is not yet seen and design cannot be influenced by this. Also, it has added a manual intervention feature to the produced form. Thus, the designer can apply a reduction to the resulting product. With such features that can be added, the application has the potential to be used in architectural design with alternatives offered by a productive system design to the designer.

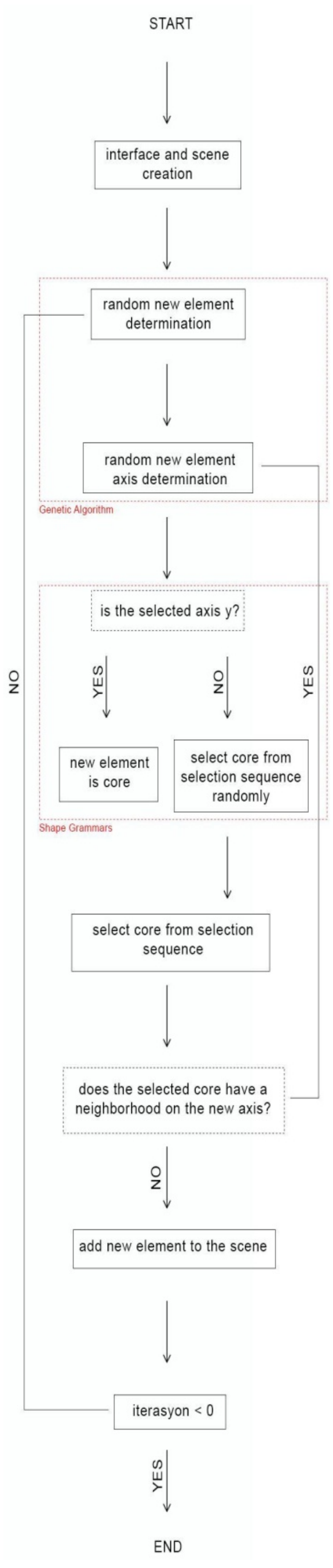

START
interface and scene creation
random new element determination
random new element axis determination
Genetic Algorithm
is the selected axis y?
YES
NO
YES
NO
new element is core
select core from selection sequence randomly
Shape Grammars
select core from selection sequence
does the selected core have a neighborhood on the new axis?
NO
add new element to the scene
iterasyon < 0
YES
END

***Figure 9.** Application algorithm flow chart.*

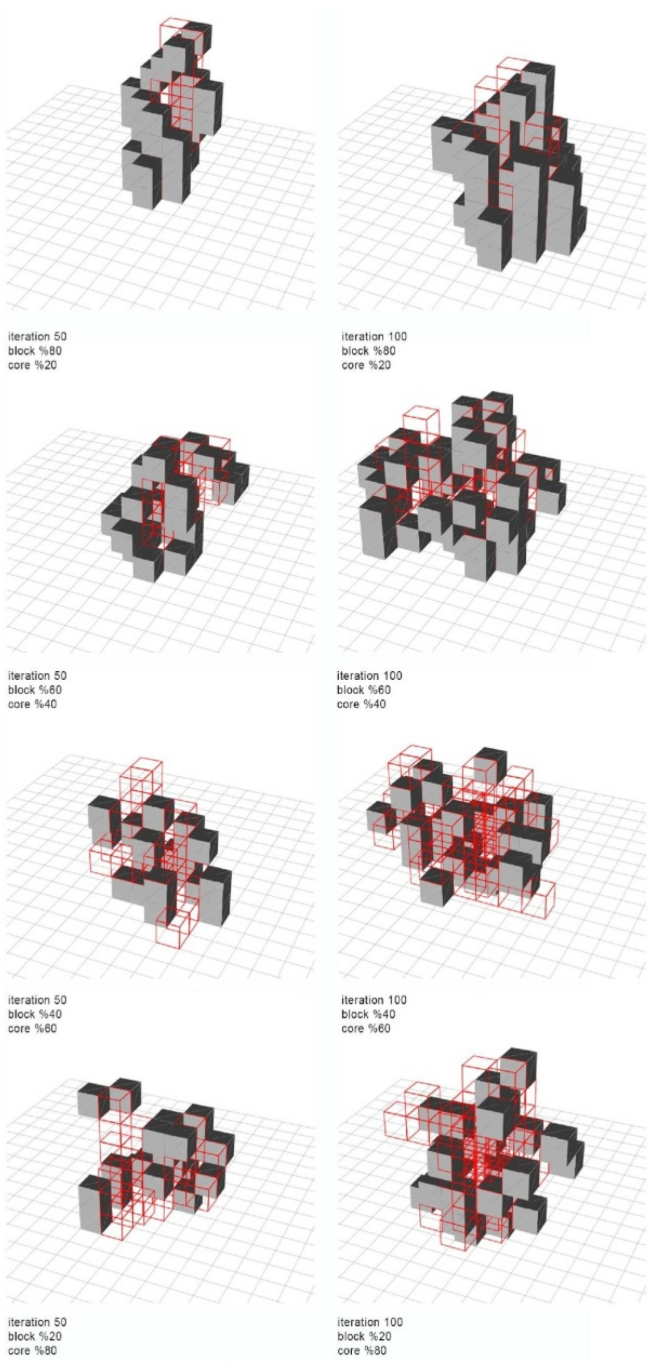


***Figure 10.** Alternatives resulting from growth with different parameters*

Future Works

Form finding has always been an important step in architectural design. In application, the major compelling point is to produce the form in space without knowledge of environment around it. So that next step of the project would try to achieve solutions to its non-contextual outcome.

Acknowledgement

We would like to specially thank Prof. Dr. Gülen Çağdaş. The early phase of this study conducted as a part of Generative Systems in Architectural Design, instructed by her, which is an Architectural Design Computing Graduate Program course in Istanbul Technical University. We would like to take this occasion to thank Prof. Dr. Leman Figen Gül for supports as our supervisor.

References

[1][4][5]Maturana, H.R. ve Varela, F.J. (1980). Autopoiesis and Cognition: The Realization of the Living.

[2]Aşkın, Z. (2010). ETHOS: Felsefe ve Toplumsal Bilimlerde Diyaloglar, Sayı: 3.

[3]Retrieved from http://sozriko.blogspot.com.tr/2015/04/aristoteles.html

[6]Diniz, N., Turner, A. Towards a Living Architecture. ACADIA 2007

[7]Hensel, M., Menges, A., Weinstock, M. (2012). Morphogenesis and Emergence.

[8][10]Maher, M.L. and Kundu, S.,(1994) Adaptive design using genetic algorithms, in J S. Gero and E. Tyugu (eds), Formal Design Methods for CAD, North-Holland, Amsterdam.

[9]Mitchell, M. (1996). An introduction to genetic algorithms. Cambridge, Mass.: MIT Press.

[11]Jones, N. (2009). Architecture as a complex adaptive system.

[12]Chouchoulas, O. (2003). Shape Evolution; An Algorithmic Method for Conceptual Architectural Design Combining Shape Grammars and Genetic Algorithms.