

Generative Algorithms in Alternative Design Exploration

M. Galieh Gunagama^{1,*}

¹Department of Architecture, Universitas Islam Indonesia, Indonesia

Abstract. The improvement of design capabilities with the help of computers has been very commonly discussed recently. Many studies have presented the topic of evaluating a condition on a design proposing a form by applying an algorithmic approach. However, this is still a discourse when it comes to design exploration at the preliminary design stage, which in general, design initiation is still dependent on humans. This research seeks to bridge the existing gap by devising an algorithm that can help create alternative designs at preliminary design stage. The case scenario presented in this study is for 1 story building on sloped land. Design limitations and site conditions are the primary considerations. The discussion focused on the dynamics of the algorithm process that was arranged for the acceleration of alternative design creations.

Keywords: Generative Algorithm, Design Exploration, Preliminary Design Stage, Architecture

Research Background

Optimization of computer capability in the design has been a dream and hope of the designer since the past four decades. Currently, design with the help of computer has become commonplace. However, the use of computers as a tool to rapidly arrange alternative designs began to be developed and not fully utilized. Previous studies have not been able to meet the expectations of a computing process that enables optimum design automation. This condition is influenced by many things, including the many variables that should be considered in the design by the architect (Lobos & Donath, 2010) and the extent of the disciplines that must be integrated into a design (Turrin, Buelow, & Stouffs, 2011)

Utilization of computer capabilities in the preliminary design stage has been done before. However, the design that using a computer in the preliminary design stage for some researchers is only seen as a mere replacement tool, which is to move the manual media into digital (Aliakseyeu, Martens, & Rauterberg, 2006) (Pranovich, 2004). Besides, simulating the design related to environmental conditions both inside and outside the building has been done before (Attia, 2011) (Attia, Gratia, Herde, & Hensen, 2012). In general, the simulation is intended to evaluate the condition of the building by predetermined standards and criteria or to find a form of building that is considered optimal in a particular domain

The development of computational design allows the design to be created freely and without limit. (Humppi, 2015) (Khabazi, 2012) So aside from being used as a tool to evaluate, should the use of computer capability does not stop at that point only. But it should be used to

encourage the design process towards creating a large number of alternatives that can speed up the preliminary design process. With the help of computer skills, the limited human cognitive power to produce alternatives should be expanded, especially as a catalyst for the creation of new possibilities (Humppi, 2015)

Although considered capable of improving human capabilities, computers are not always aware of the environment. Terzidis (2006) argues that the mental processes of design are conceived, imagined, and processed in the human mind. Thus, computers are only seen as a means of calculation, organization, productivity, and presentation. Because of that, the algorithmic results generated by computational design must refer to the human mind, i.e., programmers or designers, so that the process of criticizing the algorithm results is always directed to the designer using the algorithm. In other words, the one responsible for the final result is the designer, not the algorithm that helps generate it (Terzidis, 2006).

The discussion in this study attempts to open up the possibility of accelerating the exploration of alternative designs at preliminary design stage assisted by computational design. The absence of algorithms in computational design that refers to land with slopes and sites with vegetation encourages the idea of building algorithms that seek to solve the problems in the scenario.

This research tries to achieve that idea using Rhinoceros and Grasshopper, especially the Galapagos component. Galapagos simulates a generative process that mimics evolution to find the most optimal composition based on predetermined settings (Rutten, 2010). Although not the best method, the evolutionary solver in the Galapagos is considered capable of generating an alternative answer

* Corresponding author: galieh.gunagama@uii.ac.id

to the given problem, which in general, each new answer will be better than the previous answer (Rutten, 2010).

With these capabilities, to rapidly generate large numbers of design alternatives in the preliminary design stage, generative algorithms are considered capable to simulate the answers for the model to be built. The built model will focus on the design on land with a certain slope, with the size of the space, the existing vegetation conditions, the width of the border of the land, and the predetermined distance between building masses. This research proposes an algorithm that can be used to obtain design alternatives that have optimal value in response to site conditions

Parametric in Preliminary Design

The design process in Architecture can be divided into two stages, namely the initial design stage and final design stage. In the initial design phase, the problem has not been well defined, the thing that is done is still within the scope of exploration which includes the process of describing the sketch, schematic, and building the model. Activities undertaken are explorations in the effort to formulate design issues. Its characteristics include generating alternatives, shifting from one idea to another because of a lack of commitment to one idea, general information input, and rough visualization to reinterpret the design at different times (Pranovich, 2004).

The final design stage has the characteristics that the problem is well defined. The resulting product has been elaborated with working drawings, details, and threedimensional models. The activities that occur are elaborating solutions to the problem of space, which includes trying to formulate solutions to existing problems, sharpen ideas in more detail, including more specific information, and define ideas more precisely (Pranovich, 2004).

In other words, the initial design or preliminary design stage is the beginning when the design is conceived. This section is the starting point before the design is sharpened in the next process. In the preliminary stage, the designer will focus on how the design will be constructed both regarding partial and holistic compositions, and also the concept of beauty will be defined.

The demand to propose many alternatives to clients in this phase becomes very important and significant. Designs prepared from the best alternative will ease the next design stage for the designer. Although it is commonly known, the number of alternatives created is very limited to human capabilities. At this point the role of design with the help of computers is necessary.

This research seeks to facilitate the composition of the mass of the building at the preliminary design stage. In addition to the human's cognitive limitations, sketching, schematics, and physical models for 3dimensional studies are considered less practical and require a relatively long time compared to compiling a computer simulation. Therefore, this study seeks to enhance the design capability of the initial design stage

by proposing a model constructed using an algorithm and will represent a 3dimensional form of simulation with the ultimate goal of accelerating the creation of alternative designs at the initial design stage.

Algorithms and Design Restriction

Generative algorithms are algorithmic and parametric ways that handle geometry in design problems. With this algorithm method, designers can utilize all the capabilities and possibilities regarding geometric computing, as well as in managing large amounts of data, numbers, and calculations better than conventional geometry methods. Discussions to be conducted with this method are not limited to predetermined experiments, but to explore the unlimited potential. There are many ways to set up a design algorithm, although there is an issue of the limitations on the commands provided in the software. However, it still allows the preparation of alternative solutions that can be offered and gives unlimited possibilities (Khabazi, 2012).

Simplistically, the algorithm can be likened to a cookbook containing a step-by-step guide to making food. Maldonado (2014) argues that, as in cookbooks, recipes can be articulated as a series of interconnected steps or actions that govern the interactions between the various components. The variation of the parameters that inform the interaction potentially alters the process's final results significantly. To produce alternative dishes, substantial modifications in the process need to be done after the fundamental interactions of the given recipes are understood (Maldonado, 2014).

Therefore, if the algorithm is a system, then the variables and parameters needed to run the system. Variables and parameters can be analogous to the limitations in the design. Variables are entities that change in a particular system, whereas parameters are entities that are used to connect or unify two or more variables of an equation. Parameters may or may not be able to have the same dimensions as the variable (Difference Between Variable and Parameter, 2012).

To create a system that works well, data inputs and the boundaries need to be appropriately defined in the beginning. In architectural design, things that can be categorized as data such as land properties, including land size, land contour, existing vegetation, and climatic conditions. The design boundaries include building regulations, structural constraints, and specific values in designs that defined by the designer, such as space programming and relationships between spaces.

There are five conditions to be considered for the model made in this study. The conditions are, (1) the building is placed on sloped land, (2) the building should not be out of the boundary of the border rules of a certain size, (3) the building should not violate the preexisting vegetation growing on the site, (4) the room masses is expected not to touch each other, and (5) the ideal distance between adjacent spaces should not exceed a predetermined number.

The scenario used is a land area with a size of 35 x 20 square meters, for 1 floor building, and the building will consist of 10 rooms. All sizes in buildings and spatial information are arranged in syntactic plugins to facilitate organization and space management (Nourian, Rezvani, & Sariyildiz, 2013). In this study, Syntactic is used as a tool to define space and size only, while the built model will be evaluated using the Galapagos. The genome input to be used is the coordinate of the room located on the site and the rotation angle of the room that is modeled. While the target fitness to be addressed is to get an alternative setting for the room on the site with the optimal value of the conditions required above.

Modeling and Simulation

Modeling

The model used for the simulation is built in several stages. The first stage is the creation of the site for the simulation; the second is the preparation of room modules in the building to be simulated; the third is the modeling for the test limits; the fourth is the preparation of tests to be used.

All models are built in Grasshopper and Rhinoceros are used as visual viewers only. The Use of Grasshopper from modeling stage is done to make the produced model can be more easily modified, so it also can be applied to similar cases with different settings.

The land that will be used for simulation is a rectangular and simulated sloped area. The selection of the land arrangement in such a way is to test the model in the 2 dimensional basic shape, i.e., square on the X and Y axis, as well as at the point that has transformed to the height of the slope, i.e., on the Z axis. The selection of site simulation with this kind of form also allows the study to see which part of the room module that will sink into the ground with that setting. The model for the site shown as a surface with a contour line as a reference for its slope.

The preparation of building room module will be based on the points on the X and Y axes that refer to the length and width of the land. The reference point is arranged independently, i.e., the value of each X and Y for each point can be determined by itself. These points are then projected on the ground surface to obtain the coordinates of the Z axis that bound to the land height. The independently arranged reference point allows the plotting of room modules to be more flexible. It can also be seen as a simulation of freedom in plotting the building for the designer. The arrangement in such a way allows the occurrence of two points that have the same X and Y values. This condition is not expected to happen, so different tests are needed.

The points with the coordinates X, Y, and Z are used as the midpoint on the square which is the basic form of the room modules. The room name data and the size of the room for each module have been determined in advance. Space organizing is done with syntactic components. Part of this plugin allows space to be traced

by giving names to spaces and color sign on the display corresponding to specific room information.

Then each square for the room module is rotated with the rotation axis at the center point of reference created beforehand. The next step is to provide a height measurement for 2 dimensional room module so that the form achieved is in the 3 dimensional box. The height of the module can be adjusted with the prearranged room name data.

The rotation angle for each room module is set to be freely changeable. This allows the buildings arrangement to be not rigid on facing one side only. This condition can also be analogized as the freedom of the designer in determining the orientation of building masses.

In this simulation, 10 rooms are modeled. Variable point X, point Y, and rotation angle will be used as a genome in Galapagos. The variable for this simulation uses real numbers according to the site length and width with 1 digit after the decimal point to determine the points X and Y, and the integer value is equivalent to 180 degrees for the rotation angle.

The next stage is modeling for test. It has been mentioned that there are some limitations, i.e. (1) the building must not exceed the site boundary limit; (2) the building does not violate the preexisting vegetation on the site; (3) between room modules do not touch each other; and (4) the spacing between the room modules does not exceeds the set limit.

The limit of the border for the site is obtained by offsetting inwards for the surface of the site based on the size that has been set. In the simulations performed, the offset is the same size, so there is no difference in size between the road border and the border between the buildings. The boundary of the border that has been obtained is then given height on the Z axis and formed into a box.

This border box will be a test for the room module box. In the test, the border box will be the outermost boundary for the room module box is allowed to be on site. So, the number that is expected to appear is the maximum value for 10 room module boxes is declared within 1 border box.

The preexisting vegetation on the site is made by using a random population that produces 10 dots in the field. It is used to simulate the location of existing trees that are also random. The existing dots are then transformed into cylinders to symbolize tree trunks.

The cylinder is the closest boundary for the room module box that can be placed on the site. That is, if the room module box exceeds this limit, then the room module can touch the tree or crash into the existing tree. This is a condition that should be avoided, so the test is made to find the maximum number for 10 room module boxes to be declared on the outside of 10 cylinders of existing vegetation.

In this simulation, the room module boxes are required not to collide with each other, which means there can not be the same X and Y value. So the test is built is to find the maximum value which states that every room module boxes that exist are not colliding with another room module box. Although not colliding with each other, it does not mean that all room module

boxes must be entirely separate and unrelated. So the distance between the room module box should be set for the room that is considered interconnected. This is to simulate the proximity of distance that usually arises between interlinked spaces. Because the distance criteria between room modules should not exceed the specified limits, the modeled simulations need to add some adjustments.

The volume centroid for each room module box needs to be known. Then, the centroids for the proximity spaces are connected by a straight line. The length of the line is then entered into the test to determine whether the line length has met the specified distance limit. Thus, the test is to find the maximum value stating that each interconnecting line is within the specified range.

The distance adjustment is greatly influenced by the 3dimensional box boundary distance to the centroid, and this occurs on both tested objects. Although the gap between the boxes has entered the specified criteria, this allows a problem to happen, which is a smaller box located in the larger box. To minimize the effects of this issue, one test needs to be added to reinforce the final result.

Adjustments need to be done by including a total volume test. If the distance between the centroids is too close, the two boxes will intersect. This makes the total volume of the boxes will also be reduced compared to if the boxes stand alone. Therefore, it is necessary to limit the extent to which the total volume of the final interconnected box may be reduced compared to the original state when there has been no contact. The defined test is to find the appropriate volumetric configuration and composition, which must be greater than the minimum permitted percentage limit.

The maximum value of each test that has been done will be formulated into a mathematical equation that will be the basis of value research in fitness in the Galapagos. The value will be a benchmark whether an alternative mass building configuration is considered to meet the requirements that have been determined.

In this research, the algorithm will be tested 4 times with evolutionary solver in Galapagos to get a different alternative. Since the initial condition of the input to the genome affects the final result of the algorithm calculation, then 4 initial conditions need to be created so that the simulation can produce different alternatives. Four initial conditions are prepared by inserting the X and Y coordinate values for each building mass by configuration (1) when the X and Y values are minimum, (2) when X = maximum and Y = minimum, (3) when X = minimum and Y = maximum, and (4) when the value of X and Y is maximum. The initial condition for the building mass rotation on the simulation will be set to the minimum value of the range of provided angles.

Simulation

To know the maximum expected value of the algorithm, it is necessary to see the highest value for each test variable. The expected value for each variable is then

compiled in the equation that will cover the entire variable. The calculation result of the equation will be used as a fitness input in the Galapagos.

The simulation in Galapagos begins by defining the relation between object A and B and converting it into numerical form. The space mass box (object A) will be tested whether inside, intersect, or outside the boundary box of the site (object B). The result of a relation is encoded into 3 numeric categories, i.e., 0 if object A is inside object B, is 1 if object A intersects with object B, and is 2 if object A is outside object B. Based on that, it can be seen if all the room mass boxes are in the boundary box of the site, meaning the maximum value that must be obtained is 0.

Next is a test to see the relation of the room module box (object A) to the preexisting vegetation cylinder (object B). The value that needs to be defined is the statement "true" or "wrong" in the Boolean Operation. Since the required data is numerical, each statement needs to be converted into numbers. In Grasshopper, each false is 0, whereas each true is 1. This means that if there is an object A declared "true" collided with object B, then the value that appears is 1.

For the logic in the simulation algorithm to be in line with the predetermined requirements, the statement needs to be adjusted. The required condition is that there should be no room module box that can be tangent to the preexisting vegetation cylinder. So the corresponding statement is if there is an object A declared "not true" collided with object B, then the value that appears is 1. Based on that, if there are 10 boxes that do not collide with 10 cylinders, then the maximum value should be gained 100.

Then to test whether each box of the room collides with another box of the room, the same way with box and vegetation cases can be used. However, there is a difference in understanding the logic of the test results, especially since the input data comes from the same object. The treatment for this test result needs to be scrutinized because it will produce an output that is logically considered valid, but the value will be regarded as unsuitable.

If there is one group of objects tested whether it collides with the same group, then there will be a result for 1 piece that will always be considered "true" to collide. This is because if object A is tested to the same object, then the test value will always be "true" to collide with each other. Logically, this is considered valid because it means that object A resides within object A. Similar results will also occur when object B is tested with object B, object C with object C, and so on. But it will not happen if the object is different like when object A is tested with object B, C, and so on. Thus, for minimum value search, this needs to be an additional calculation, i.e., to eliminate the test result value between the same object as the exclusion criterion. Otherwise, it can capture the maximum value for tests on different objects as inclusion criteria.

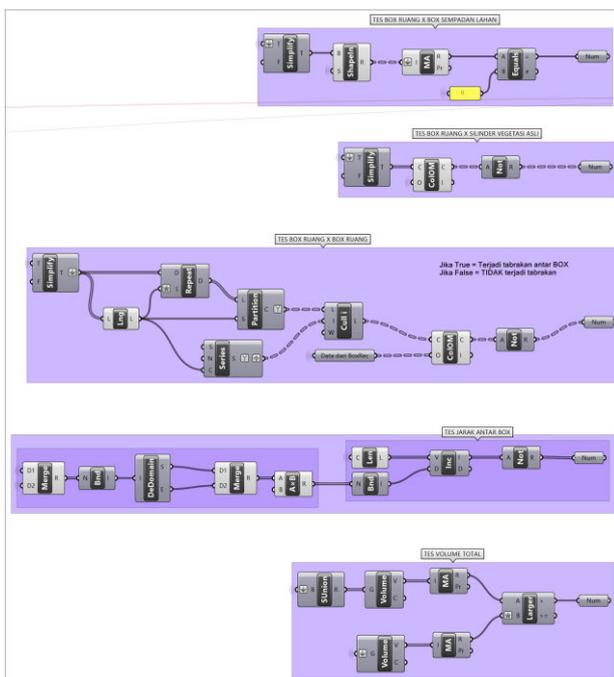
Based on that, it can be seen if 10 objects are tested with the same 10 objects, then the result will always be 10 test results that will always "true" collide with itself and 90 other test results that have not been determined as

"true" or "false." Therefore, in the calculation to find the maximum value, the value of 10 for the "true" test result will be eliminated, and the expected maximum value is 90 for the "incorrect" statement of the building module box colliding with each other.

To evaluate the distance between the boxes, the test used is different from the other parts. Instead of using a 10 point building module, in this test using 11 points is to add 1 line that determines the entrance area into the site. This one point will simulate the closest distance from the site entrance to the earliest building module encountered with circulation, i.e., carport or garage. This is a consideration because the site alone without any circulation is considered unlikely to happen. Therefore, if the line length is tested for a specific range and if it is entirely "true" then the maximum value is 11.

As for the evaluation of the total mass volume of the building, the test will be made to compare the sum of the original volume of the entire mass of the building and the volume when it has come into contact. In Boolean Operations, merging or unification (union) of multiple solid objects may occur. This allows the changes in both surface area and overall volume.

The changed volume is used as the tested variable, while 90% of the original volume is the reference variable. Determining the percentage of the initial volume is an attempt to simulate the degree of freedom in creating the composition of the building mass, and to what extent the percentage of space can change compared to the original size. The use of this system in addition to aims to separate between one object with another also provides an opportunity on the existing object in order to be tangent to simulate freedom in designing. The calculation result in this test will only take 1 "right" condition that is when the volume of the object after tangent is higher than 90% of the original volume.



Fitness

Khabazi (2012) argues that the idea of finding solutions or optimizing products based on design problems that have been set is fascinating for various reasons. It may encourage the designer to think through the logic of each designed product, to formalize this logic with the language of mathematics and geometry, and to encourage constant movement toward better solutions by adding more criteria to evaluate a product (Khabazi, 2012).

The process of the algorithm making in this study also experienced a similar thing. For the final aim to be achieved, additional criteria and updates are done each time the algorithm is tested on the model. The results in the initial stage simulation were analyzed to find the weakness and improved the test in the next opportunity. The process continues to be repeated until the algorithm gets closer to the expected condition.

Based on the process mentioned earlier, the results of each test will be symbolized in the variable and arranged in an equation that will be used to assess fitness. Therefore, the symbol for the test results of the room module box for the bounding box = a; test result of room module box to cylinder of preexisting vegetation = b; test results of room module box toward the same room module box = c; test result of evaluation distance between boxes = d; and test result for total volume of the boxes = e.

In the compiled algorithm, the evolutionary solver is set to find the maximum value so that some values of the variables need to be adjusted in their fitness equations. Because variable a will be minimized and is a prerequisite that must be fulfilled, so it is placed as a multiplier. The test results of variable a must be = 0, which means that none of the space module boxes violate the border box. If this can be met, then it can be defined as true with value = 1. Thus, as long as the value of a is still not reached or false, the fitness value will be ignored because of the final result of calculation = 0. In such conditions, the solver will repeat the simulation and automatically search a better alternative fitness value.

The variable values b and c are the values that will be maximized, but still, allow the existence of the intersection between the space module boxes with the vegetation cylinders and the module boxes with itself. So both are used as the search for the optimal value to be divided. However, the value of these two variables will only be calculated if the value of variable e as multiplier get the maximum value, which is true. Same as variable a, as long as the value of e is still false, then the equation will be locked at 0, and the fitness result is deemed ineligible. This will force the solver to continue the search for better fitness value.

The variable d will act as a divider on this equation. Although the required value d is the maximum, that is all the distance will be true if it is at a predetermined number, but still allowed for any value that does not match the requirements. This value is adjusted to get the opposite value described above, then squared. This is done so that if there is only 1 value that false in the variable d, it will decrease the final result. That is, the

true value is still maximized by minimizing the false value.

The number 1 is the constant required for the equation to be calculated. If the value of $d = 0$, which means the calculation is successfully maximized, and that value becomes the only divider, then the whole fitness will be an error because all that is divided by 0 yields undefined value. To avoid this, the number 1 as constant will be the only divider in that scenario, which allows fitness to be calculated.

Based on the formula, the ideal value obtained is 190 with the ideal value for variable $b = 100$ and variable $c = 90$. While the other variables because it works on Boolean Operation as the multiplier (minimum variables $a = 1$ and $e = 1$), or work to find the minimum value ($d = 0$), so the resulting value does not appear in the final result, but it will determine fitness success. The simulation will be stopped after it is considered close to the ideal number and before passing through the 140th generation. This is to shorten the time as the primary purpose of this research, which is to accelerate the creation of alternative designs.

In the first simulation with initial conditions both X and Y minimum, up to 3rd generation has obtained good condition with fitness score 41.0. After gaining a score of 164.0 in the 19th generation, the scores for the given alternatives rose gradually until the 102nd generation began to stagnate at a score of 185.0. In the 139th generation, the simulation was stopped because the fitness value was considered stagnant and no longer produced a new alternative.

In the second simulation with initial condition X = maximum and Y = minimum, what happens is up to 10th generation simulation has not found suitable condition with fitness score only 6.96. However, after a score of 171.0 in the 11th generation, the scores for the given alternatives improved considerably until the 76th generation began to stagnate at a score of 186.0. In the 105th generation, the simulation was stopped because the fitness score had stagnated and no new alternatives were completely different.

The third simulation starts with initial condition X = minimum and Y = maximum. In the 5th generation, the algorithm has been able to produce a soaring fitness score that is from 39.5 to 167.0. The score gradually increased until the 73rd generation got 186.0 and after that, the scores for the given alternatives began to stagnate. The simulation was stopped in the 111th generation when completely different new alternatives were not found.

The initial conditions for the fourth simulation are both X and Y maximum. The simulations show that up to the 32nd generation fitness scores are only at 6.96 and just 42.0 in the 33rd generation. This score goes up gradually until the 71st generation to 45.0, which then increased sharply to 178.0 in the 72nd generation. This score is improved but not significantly until the simulation is stopped because the algorithm does not provide a different alternative, i.e., in the 121st generation with a fitness score of 184.0.

In the four simulations, with different initial conditions, the working process also seems different.

Some simulations seem quite quick to provide solutions with high scores in the early generations, some of which need up to tens of generations to score high. It occurs in addition to being affected by the initial conditions, also influenced by the way the evolutionary solver works.

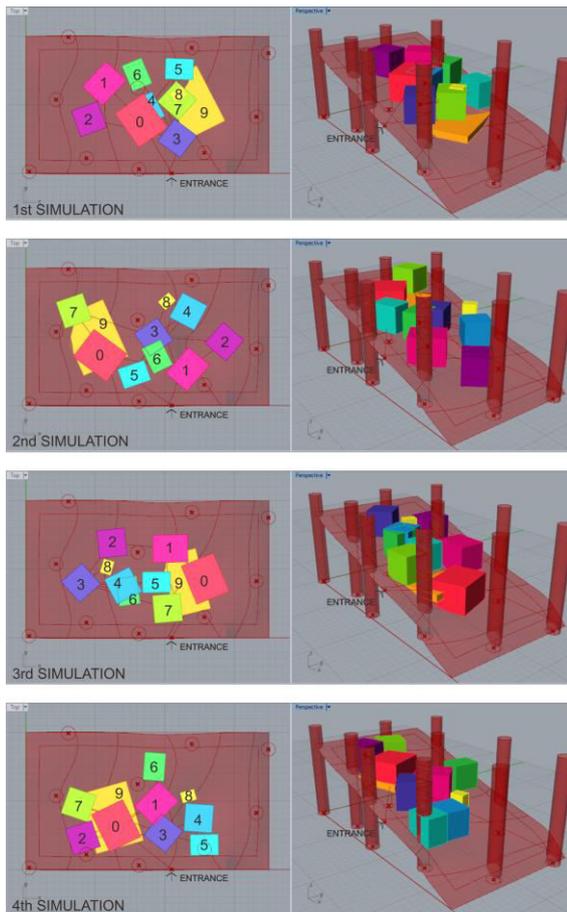
As mentioned in the previous section, the evolutionary solver is an imitation of the evolutionary process (Rutten, 2010), i.e., the probability of a generation resulting from the parent gene pair. Simulations that have a low score up to several generations occur because the solver seems to be fascinated with a gene alternative with low value and does not seem to be able to switch to another possible gene that is unusual, although it may be better. The composition of the gene with the low value appears to be the best, even though it is not ideal yet. This kind of simulation condition will last until found better value.

The opposite condition also occurs when the early generations have produced high fitness scores. The solver will automatically eliminate genes that produce low value generations. Because high values are preferred, so in the next generation solver will work only on selected combinations of genes only. For the long term, the same score for several generations will be considered redundant, and fitness will appear stagnant. This is a weakness of the evolutionary solver as Rutten (2010) has disclosed. Nevertheless, the ability of this solver in the creation of design alternatives quickly is the intended point in this study. The results obtained in this study may be different from other reviews because of the initial conditions and the nature of the solver that indeed allows differentiation to occur

Discussion on Alternative Design

The design alternatives generated from the algorithm are quite varied. From 4 initial conditions, 4 alternatives were selected which had the highest score and were in the last generation before the simulation was stopped. The four alternatives obtained have their advantages and disadvantages that can stimulate a better idea at a later stage.

Like an alternative is offered from the first simulation. The pool area is on the lowest ground and directly adjacent to the back patio area which may encourage discussion to maximize visual to the lower side of the land. However, in this simulation plotting of the front porch and lounge areas looks unusual and separate from other buildings.



0: Carport; 1: Front Porch; 2: Lounge; 3: Family Room; 4: Dining Room; 5: Kitchen; 6: Main Bedroom; 7: Backside Porch; 8: Bathroom; 9: Swimming Pool

While on the second alternative, the location of the kitchen that prevents direct circulation from the entrance area to the carport can be considered less suitable. Although the mass of the building looks more separated than other alternatives, the location of the family room in the middle of building masses can be seen as a binder between several building masses.

The third alternative design allows discussion of design ideas by lowering the carport area and putting the pool in a higher position beside the carport. So get additional open area above the carport that can be utilized for the activity. Meanwhile, the kitchen area is the link between the swimming pool, the two terraces and the dining area. However, the location of the lounge and living room are quite separate from other buildings, and the combined dining and bedroom rooms may not be so desirable that they can be fixed again at the next design stage.

Meanwhile, the fourth alternative design allows the location of the bedroom that can maximize the atmosphere of a quiet room. The location of the unobstructed bedroom with another building enables visual optimization towards the lower site. However, placing the carport piled with swimming pools and adjacent to the two terraces can be considered unusual and less expected.

At a glance, design alternatives offered can be called meaningless because of various things. For example, through a more conservative perspective, the space that should be able to connect directly, but on the

alternatives offered is not interconnected, can be regarded as a bad design. Besides, discussions about spaces that are displayed separately in their respective boxes without visual attachment as a building may complicate security control or in organizing utilities. Also, the simulated display in the form of a box can give rise to a very rigid design impression and cannot accommodate other geometric shapes.

In fact, alternative designs at the initial design stage can also be translated as a guide for developing the model at a later stage. The discourse is primarily about the ability of the algorithm to explore the ideas of arranging the composition at the initial design stage. So the alternative can be used as a discussion and consideration. Then, from this point on, the designer can enter into further design discussions to interpret and make adjustments so that the design results can be accepted.

Conclusions and Recommendations

Based on the explanation in the previous section, generative algorithms are considered capable of producing various alternatives. However, due to the breadth of the optimal design definition and the lack of ability to translate verbal ideas to mathematical, the resulting alternatives are still limited to the minimal. The addition of complementary variables and the deepening of the effort to convert the ideas into mathematical notation will be a good starting point for subsequent research.

Because of the lack of variables mentioned above, the design aesthetics produced are also very likely to be discussed and further criticized. The proposed alternative can be considered reasonable and rational for some designers, and otherwise can even be regarded as fanciful and unusual for some others. Although opening an experimental discussion may open up the possibility of receiving the alternative, the issue of aesthetic perceptions gained from this simulation remains a significant concern that needs to be followed up.

The algorithm that is built is still on a simple level. This can be seen from the building used as a case study, i.e., a standard 1-storey house on sloped land. Because this research tries to generate design alternatives with an algorithmic approach in such cases, the discussion limits do not include the expansion of more complex alternative possibilities, such as the multi-story buildings or buildings with more mass. Hence, discussions to improve the ability of algorithms with higher levels of complexity can be done on other occasions.

The high number of genomes that became input on the simulation in this study may allow infinite design alternative. However, fitness designed to be strict and have a clear direction becomes important in finding the expected design alternative. Nevertheless, both of these results in slower computer calculating performance.

Preparation of models and simulations in Grasshopper is considered more flexible regarding modification. Of course, the discussion of the role of a designer in computational design can be more referring

to two things, namely the designer as a human who builds algorithms and designers as humans who interpret alternative design results. This condition is to emphasize that computational design is the result of the human designer who in the process is assisted by the machine. This process aims to accelerate the creation of alternatives given primarily at the initial design stage.

Turrin, M., Buelow, P. V., & Stouffs, R. (2011). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. *Advanced Engineering Informatics*, 25(4), 656–675.

References

- Aliakseyeu, D., Martens, J., & Rauterberg, M. (2006). A computer support tool for the early stages of architectural design. *Interacting with Computers* 18, 4, 528-55.
- Attia, S. (2011). *State of the Art of Existing Early Design Simulation Tools for Net Zero Energy Buildings : A Comparison of Ten Tools*. Louvain La Neuve, Belgium: Architecture et climat, Université catholique de Louvain.
- Attia, S., Gratia, E., Herde, A. D., & Hensen, J. L. (2012). *Simulation-Based Decision Support Tool for Early Stages of Zero-Energy Building Design*. *Energy and Buildings*, 49. doi:10.1016/j.enbuild.2012.01.028
- Difference Between Variable and Parameter. (2012, Mei 31). (DifferenceBetween.Com) Retrieved September 29, 2017, from <http://www.differencebetween.com/difference-between-variable-and-vs-parameter/>
- Humppi, H. (2015). *Algorithm-Aided Building Information Modelling; Connecting Algorithm-Aided Design and Object-Oriented Design*. Tampere University of Technology.
- Khabazi, Z. (2012). *Generative Algorithms. Morphogenesisism*.
- Lobos, D., & Donath, D. (2010). The problem of space layout in architecture: A survey and reflections. *Arquitetura Revista*, 6, 136–161.
- Maldonado, M. P. (2014). *Digital Recipes; A diagrammatic approach to digital design methodologies in undergraduate architecture studios*. eCAADe. 1. Fusion-Data Integration at its best.
- Nourian, P., Rezvani, S., & Sariyildiz, S. (2013). *A SYNTACTIC ARCHITECTURAL DESIGN METHODOLOGY : Integrating real-time space syntax analysis in a configurative architectural design process*. Ninth International Space Syntax Symposium.
- Pranovich, S. (2004). *Structural sketcher : a tool for supporting architects in early design*. doi:10.6100/IR576067
- Rutten, D. (2010). *Evolutionary principles applied to problem solving*. Retrieved Januari 17, 2017, from Grasshopper3D: <http://www.grasshopper3d.com/profiles/blogs/evolutionary-principle>
- Terzidis, K. (2006). *Algorithmic Architecture*. Burlington, MA.: Architectural Press.