GENERATIVE DESIGN FOR SPATIAL LAYOUT OF URBAN NEIGHBORHOOD

A DISSERTATION

Submitted in partial fulfilment of

the requirements for the award of the degree

of

MASTER OF ARCHITECTURE

by

ABHISHEK PALIT

(Enrollment No.: 19510002)

Under the Guidance of

Prof. Saptarshi Kolay

DEPARTMENT OF ARCHITECTURE AND PLANNING

INDIAN INSTITUTE OD TECHNOLOGY, ROORKEE

JUNE 2021

CERTIFICATE

Certified that dissertation entitled “**GENERATIVE DESIGN FOR SPATIAL LAYOUT OF URBAN NEIGHBORHOOD**” which has been submitted by **MR. ABHISHEK PALIT**, in partial fulfilment of the requirements for the award of the postgraduate degree of Master in Architecture, in the Department of Architecture and Planning, Indian Institute of Technology Roorkee (IITR), is the student’s own work carried out by him under my supervision and guidance. The matter embodied in this dissertation report has not been submitter for the award of any degree of this or any other institute.

Guided by

Saptarshi Kolay,

Assistant Professor.

Department of Architecture & Planning,

Indian Institute of Technology Roorkee,

Roorkee, Uttarakhand, India – 247667

Date: June 2021

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"GENERATIVE DESIGN FOR SPATIAL LAYOUT OF URBAN NEIGHBORHOOD"** in the partial fulfillment of the requirement for the award of the degree of Master in Architecture, submitted to the Department of Architecture and Planning, Indian Institute of Technology- Roorkee, is the authentic record of my own work carried out during the period from July 2020 to June 2021 under the guidance of Saptarshi Kolay, Assistant Professor, Department of Architecture and Planning, Indian Institute of Technology – Roorkee, India.

The matter embodied in this report has not been submitted by me for the award of any degree of this or any other institute.

Date: June 2021

Place: Roorkee

Abhishek Palit

M.Arch 2nd Yr

Enrollment no. 19510002

This is to certify that the above statement made by the candidate **Mr. Abhishek Palit** is true to the best of my knowledge.

Saptarshi Kolay,

Assistant Professor.

Department of Architecture & Planning,

Indian Institute of Technology Roorkee,

Roorkee, Uttarakhand, India – 247667

Date: June 2021

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my guide Prof. Saptarshi Kolay for the continuous support in my Dissertation study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing this dissertation. I could not have imagined having a better advisor and mentor for my Dissertation and study.

Besides my guide, I would like to thank my friends Navneet Kumar and Rajon Debonath and Vidisha Pagariya, for the stimulating discussions that helped me develop my thesis and idealize new innovative ideas. Also, I would like to thank Kean Welmsley of Autodesk for his advice on the development of Generative Algorithm.

Last but not the least, I would like to thank my family, my parents Uday Palit and Baijayanti Palit for guiding and supporting me spiritually. Also, my sisters, Ankita Palit, Akriti Palit and Niyati Palit for always listening to my ideas and encouraging me always.

Abstract

The Global population increase coupled with increasing urbanization has increased the pressure on urban areas and has also emphasized the magnitude of challenges faced by city planners and architects. The increasing demand of land in urban areas is leading to the extension of city boundaries and development of new neighborhoods. These new neighborhoods need to have an optimized layouts for the overall wellbeing of the residents, through informed urban design.

Urban design is a complex process that involves multiple parameters and goals and include multiple stakeholders. Often, the city planners and architects find it challenging to find the right design that incorporates all the different goals of all stakeholders.

The dissertation is focused on creation of a Generative Algorithm that provides multi variate scenario of the neighborhood based on certain specified set of goals and parameters. The study is done for the Block BD and CD of the Salt Lake City neighborhood in Kolkata. To identify the design success, different design goals were selected which included (a.) Profit – By maximizing no. of Plots and minimizing the number of Roads; (b.) Urban quality – By maximizing the Open Spaces; (c.) Proximity – By minimizing the distances to School and Open Spaces; (d.) Urban comfort – By maximizing the shading percentage of outdoor spaces.

The Generative design algorithm has been developed in the dissertation and has been tested to output multiple design solutions. A normalized metric has been developed to assess the different design solutions based on their goal specific numbers. Ten high performing design solutions have been selected in the study based on their normalized metrics. Two best performing solutions have been compared to the existing neighborhood design. It is found that the solutions generated by Generative Algorithm score higher that the existing design of the neighborhood.

This page is intentionally left blank

Table of Contents

1. Synopsis 15
1.1 Introduction 15
1.2 Need of the Study 16
1.3 Aim 16
1.4 Objectives 17
1.5 Scope 17
1.6 Limitations 17
1.7 Expected Output/Deliverable 17
1.8 Research Methodology 18
2. Literature Review 20
2.1 Introduction 20
2.2 Background 20
2.3 Generative Design 22
2.4 Multi Objective Optimization 26
2.5 Genetic Algorithms 29
2.6 Non-Dominated Sorting Genetic Algorithm II (NSGA – II) 31
2.6.1 Fast Nondominated Sorting 31
2.6.2 Crowding Distance Computation 34
2.6.3 Main loop 35
2.6.4 Tournament selection 36
2.6.5 Crossover and Mutation 36
2.6.6 Summary 39
2.7 Case Study 40

2.7.1 Case Study 1 – Computational Urban Design Prototyping: Interactive Planning Synthesis Methods – A case study in Cape Town. 40

2.7.2 Case Study 2 – Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout 51

3. Site Analysis 58

3.1 Site Introduction – Salt Lake City, Kolkata 58

3.2 Location of Blocks CD and BD – Study area 60

3.3 Figure Ground Map 62

3.4 Land – Use Map 63

3.5 Block Spatial Design Analysis 64

3.6 Amenities 67

3.7 Shading Analysis 69

3.8 Summary 70

4. Project Development 71

4.1 Parametric Model 73

4.1.1 Site Boundary 74

4.1.2 Primary Roads – Through streets 74

4.1.3 Major Plots 76

4.1.4 Secondary Roads – Local streets 77

4.1.5 Secondary Plots 80

4.1.6 Amenities 81

4.1.7 Building Blocks 84

4.2 Input Parameters 85

4.3 Design goals 86

4.3.1 Profit: Maximizing Number of Plots 86

4.3.2 Maximizing Percentage Open Space Area 86

4.3.3 Minimizing Road Area Percentage 87
4.3.4 Maximizing Proximity to School 87
4.3.5 Maximizing Proximity to Open Spaces 88
4.3.6 Maximizing Percentage of Road Shaded.. 90
4.4 Optimization study – Generative Design Study Types 92
5. Implementation.. 93
5.1 Generate 95
5.2 Evaluation of Generated Design Solutions.. 96
5.2.1 Normalized Metrics 96
5.2.2 Analysis Graph 101
5.3 Presentation of Results 102
5.4 Results 105
5.4.1 Goal specific highest performing design solutions. 105
5.4.2 Average Performing Solutions: 111
5.4.3 Summary of selected Solutions 115
5.5 Comparative Analysis of Existing Design with Generative Design Solutions . 116
6. Conclusion 120
7. References... 125

Table of Figures

Figure 1: The evolutionary process in nature (Darwin's finches or Galapagos finches. Darwin, 1845) 23
Figure 2: The process of evolution in the nature. (Nagy, 2017) 23
Figure 3: The Generative Design workflow (Walmsley & Villaggi, 2018) 24
Figure 4: Three different configurations for the façade (Rohrmann, 2019) 25
Figure 5: The function of a parametric model. (Rohrmann, 2019) 26
Figure 6: The Graph represents the Pareto frontier formed by solutions A and B. The solution C does not lie on Pareto frontier being dominated by A and B. 28
Figure 7: Basic workflow of a Genetic Algorithm (GA) (Nagy D. , Learning from nature, 2017a) 29
Figure 8: Here, no solution dominates A; $n_p = 0$. 31
Figure 9: The domination count of B is 2 as it is dominated by A and F. 32
Figure 10: Here, all the domination scores and sets are represented, for each solution. 32
Figure 11: Code for NSGA-II Algorithm. (Agarwal & Et. Al. , 2002) 32
Figure 12: Domination score reduced to find the next Pareto Frontier group. 33
Figure 13: Domination score again reduced by 1 to group the third pareto frontier. 33
Figure 14: Final Sorting of the Solutions. 33
Figure 15: Calculation for Crowding Distance. 34
Figure 16: NSGA-II Workflow 35
Figure 17: Single Point Crossover. 36
Figure 18: Example of a uniform crossover. 37
Figure 19: Flipping Method for Mutation 38
Figure 20: Interchanging Method for Mutation. 38
Figure 21: Reversing Method of Mutation. 38
Figure 22: Workflow of a NSGA-II Algorithm. (Agarwal & Et. Al. , 2002) 39
Figure 23: CPlan Software Interface 42
Figure 24: Street Network Instruction Tree 43
Figure 25: Land Parcel representation as slicing tree. 43
Figure 26: Method of land parcellation using slicing tree. 44
Figure 27:Bounding Box for irregular site geometry. 44
Figure 28: Updated Slicing Tree data structure to include edge identity. 45
Figure 29: Crossover process for Evolutionary Multi Objective Optimization. 46
Figure 30: Study Area - Enkanini in Cape Town City, South Africa 47
Figure 31: Urban Layouts generated by the algorithm. Left: Without any specified street segment. Right: With specified Street segment. 48
Figure 32: Adding building mass to the generated output. 48
Figure 33: Empower Shack Project for online viewing and interaction with the model. 49
Figure 34: Pedestrian Accessibility Map. 50
Figure 35: Parametric model description for generating each design option 52

Figure 36: Description of two metrics, development profitability (a) and potential for solar gain (b). 53
Figure 37: Relationship and trade off goals at performance extreme 54
Figure 38: Design space visualization with plot x and y axes representing input parameters and z-axis and color representing averaged values of output metrics 55
Figure 39: Plot showing trade-off between the two objectives & 3 chosen high performing design 56
Figure 40: Location of Salt Lake City 59
Figure 41: Map of Salt Lake City, Kolkata with study area marked. (Block BD and CD) 59
Figure 42: Sector Map of Salt Lake City showing Block BD and CD. 60
Figure 43: Satellite Map of Block BD and CD. 61
Figure 44: Figure Ground Map 62
Figure 45: Land Use Map 63
Figure 46: Block dimensions 64
Figure 47: Neighborhood Road network 65
Figure 48: Secondary Block dimensions 66
Figure 49: Amenities Location 67
Figure 50: Amenities Location 68
Figure 51: Site Shadow Analysis 69
Figure 52: Methodology for Generative design Algorithm 72
Figure 53: Getting Site Boundary 74
Figure 54: Major Road Location Points 75
Figure 55: Creating parametric Road center lines 75
Figure 56: Adding Road Width 76
Figure 57: Splitting Site Geometry with Road Surface to create Major Plots. 76
Figure 58: Major Plot division based on User defined input. 77
Figure 59: Local Street Offset Calculation 77
Figure 60: Local Street Center lines. 78
Figure 61: Adding Local Street Width 78
Figure 62: All Roads defined. 79
Figure 63:Road Variation 1 79
Figure 64: Road Variation 2 80
Figure 65: Creating Secondary Plots. 80
Figure 66: Finding closes secondary plot to the center of the site to locate the school. 81
Figure 67: Closest Plot to center of Site 81
Figure 68: Defining School area and creating school building. 82
Figure 69: Finding center points of major plots 82
Figure 70: Finding Closest Secondary plot to center of Major Plots 83
Figure 71: Garden Plots 83
Figure 72: Final Neighborhood Parametric Model with building blocks. 84
Figure 73: Three versions of the Output with different parameter values of Number of Roads and Location of Roads. 84
Figure 74: Variable Parameters for the Generative Algorithm 85

Figure 75: Design goal 1 - Maximizing Number of Plots.. 86
Figure 76: Design Goal 2 - Maximizing Open Space Area 86
Figure 77: Design Goal 3 - Minimizing Road Area Percentage 87
Figure 78: Finding center point of each individual plots and school. 87
Figure 79: Finding radial distance of each plot to the school... 88
Figure 80: Finding the maximum radial plot distance to school. 88
Figure 81: Finding center points of open spaces and all individual plots 89
Figure 82: Finding radial distance of plots to open spaces... 89
Figure 83: Finding longest distance to an open space. 89
Figure 84: Finding Solar Angle 90
Figure 85: Shadow on road surface 90
Figure 86: Finding the shading angle and the shadow casting on the road.................. 91
Figure 87: Calculating percentage road area shaded. 91
Figure 88: Randomize Study method 92
Figure 89: Cross Product Method 92
Figure 90: Like this Method 92
Figure 91: Workflow of the Generative design study 94
Figure 92: Workflow of the Generative Design algorithm 95
Figure 93: Normalized metric for No. of plots 96
Figure 94: Normalized metric for Open space area % 97
Figure 95: Spider Graph Representation for generated outcomes. 101
Figure 96: Generative Design study parameters 102
Figure 97: Generated Design options. From top to last - Ascending order of open space size. 103
Figure 98: Scatter Plot 104
Figure 99: Parallel Coordinates Plot 104
Figure 100: Solution 1 - Radar Chart 105
Figure 101: Solution 1 - Outputs and Variables 105
Figure 102: Solution 1 - Design solution with maximum no. of Plots. 105
Figure 103: Solution 2- Radar chart 106
Figure 104: Solution 2 - Outputs and Variables 106
Figure 105: Solution 2 - Maximum Open area % 106
Figure 106: Solution 3 - Radar chart 107
Figure 107: Solution 3 - Outputs and Variables 107
Figure 108: Solution 3 - Minimum Road Area Percentage 107
Figure 109: Solution 4 - Radar Chart 108
Figure 110: Solution 4 - Outputs and Variables 108
Figure 111: Maximum Proximity to School 108
Figure 112: Solution 5 - Outputs and Variables 109
Figure 113: Solution 5 - Maximum Proximity to Open Space 109
Figure 114: Solution 6: Radar Chart 110
Figure 115: Solution 6 - Outputs and Variables 110
Figure 116: Solution 6 - Maximum Shading Area 110

Figure 117: Solution 7 - Radar Chart.. 111
Figure 118: Solution 7 - Outputs and Variables.. 111
Figure 119: Solution 7 - 3D View.. 111
Figure 120: Solution 8 Radar Chart... 112
Figure 121: Solution 8 - Outputs and variables 112
Figure 122: Solution 8 - 3D View.. 112
Figure 123: Solution 9 - Radar Chart... 113
Figure 124: Solution 9 - Outputs and Variables.. 113
Figure 125: Solution 9 - 3D View.. 113
Figure 126: Solution 10 - Radar chart 114
Figure 127: Solution 10 - Outputs and variables 114
Figure 128: Solution 10 - 3D View.. 114
Figure 129: Existing Layout of the BD and CD Block Neighborhood........................... 116
Figure 130: Solution 5 117
Figure 131: Solution 3 118

This page is intentionally left blank.

1. Synopsis

1.1 Introduction

Generative design is defined as "a framework for combining digital computation and human creativity to achieve results that would not otherwise be possible. It involves the integration of a rule-based geometric system, a series of measurable goals, and a system for automatically generating, evaluating, and evolving an exceptionally large number of design options" (Walmsley & Villaggi, 2018). Generative design process includes parametric design software for modelling of space, simulation software for evaluation of design options and optimization solvers for finding the most optimal design. The goals and metrics of the requirement is set by the user for a specified design problem. Based on the rule algorithm, the computer interprets the requirement and generates multiple design options as output which can be assessed by the designer.

Urban design is a complex process that involves multiple parameters such as by-laws and policies, energy optimization, environmental considerations, social factors, design factors, built-uses, proximity, and so on which influence the urban fabric and the quality of living of the inhabitants. Integrating all the different parameters and deriving the optimum design solution is a difficult and complex process. The traditional design approach relies on the intuition and previous experience of the designers which can limit the potential for novel design solutions.

The research intends to identify the different parameters and their set of required goals for urban design and spatial layout of a neighborhood and create an algorithm for automated design layouts. These parameters can be quantified through mathematical expressions and rule based geometrical system. The quantified parameters can be used to create a generative algorithm that outputs different automated design layouts of the neighborhood. These design options can then be assessed for the optimal solutions. These design solutions can thus be compared to the existing layout of the sites to identify the strengths and weaknesses of the computer-generated designs solutions.

1.2 Need of the Study

The global population is increasing at a tremendous rate, with 2.5 billion people in the 1950s to 7.7 billion people in 2019, which is expected to increase to 9.3 billion by 2050 (UN, 2020). The level of urbanization is also growing at a tremendous rate, presently 55% of the global population is living in cities, which is expected to increase to 68% by 2050. With the significant population increase occurring in cities, urban areas will be expected to accommodate an additional 2.5 billion people in the next 30 years, with close to 90% of the increase in Asian and African countries (UN DESA, 2018).

The increasing urbanization has led to high land demand in the urban areas, which is resulting in urban sprawl and haphazard development with less focus on the urban fabric. There is a need for Sustainable measures for Urban Design and planning, which is also highlighted by the UN in SDG Goal 11. Urban planning and design are a complex mechanism that involves multiple parameters and goals which benefit different stakeholders in a sustainable and equitable approach. There is a need for a comprehensive approach that inculcates all different parameters and pushes design boundaries to achieve the various goals and satisfy the needs of different stakeholders.

The growing pressure on urban areas also emphasizes the magnitude of challenges faced by city planners and architects. The current methods do not adequately accommodate the magnitude of variables that influence the urban fabric. There is a need for robust tools and techniques that would benefit urban planners and architects.

1.3 Aim

To create a generative design methodology for urban design and spatial layout of neighborhood that incorporates different variables and parameters as a set of defined measurable goals and generates different design options that can be evaluated to achieve the optimum solution.

1.4 Objectives

1. Literature Review of the Generative Design Approach.
2. Site Study and Issues Identification.
3. Identification of parameters and goals and creation of a generative algorithm that integrates a rule-based geometric system and a series of measurable goals and parameters and generates multi variate scenarios.
4. To evaluate the generated scenarios and select the optimum design option.

1.5 Scope

1. Includes the scale of a neighborhood in two selected cities.
2. The tools for the computation will include parametric software Dynamo and Generative Design Tool in dynamo.
3. Computational algorithm thus generated would output automated layouts of the neighborhood for the selected study area.
4. The study will also compare the computer-generated layouts to the existing layout of the neighborhoods of the study area.

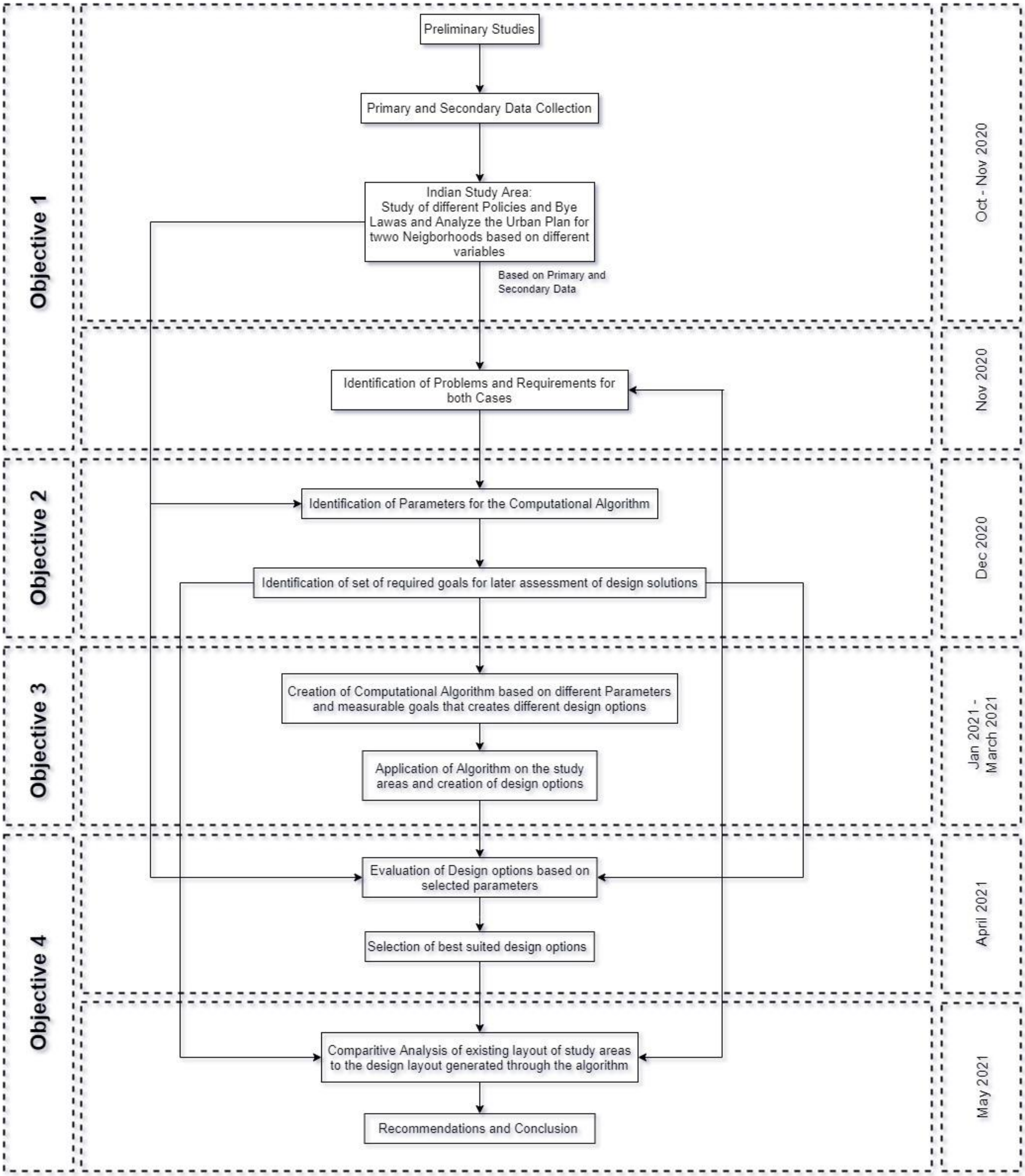
1.6 Limitations

1. The study is limited to a general statement by one city.
2. The generative algorithm will be based on the parameters and goals majorly identified through the analysis of the study area.

1.7 Expected Output/Deliverable

1. Analysis of Neighborhoods based on certain set of Parameters
2. A Generative Algorithm that gives automated spatial layout of Urban Neighborhood (based on certain selected parameters)
3. Design Solutions generated by the algorithm for the neighborhood.
4. Analysis and Comparison of computer-generated design solutions.
5. The study will enhance the research towards the use of Artificial Intelligence through Generative Design in Urban Space Design Applications.

1.8 Research Methodology



This page is intentionally left blank.

2. Literature Review

2.1 Introduction

Urban design is a complex process that involves multiple parameters such as policies and byelaws, energy optimization, environmental considerations, social factors, design factors, built-uses, proximity, and so on which influence the urban fabric and the quality of living of the inhabitants. The increasing complexity of contemporary urban design has proved the traditional design methods to be obsolete. To address the multitude of environmental, organizational, and societal challenges, urban designers require new methods and tools that integrates a multi-dimensional approach towards Urban design problem solving.

The traditional Urban design approaches relies on the intuition and previous experience of the designers who provide limited number of design options, inefficient in incorporating all the design parameters and goals, and these designs are implemented without testing and evaluation. To design sustainable, resilient, and livable urban environments, iterative testing and evaluation of design options that optimize the magnitude of site variables is required to be included in the design process.

The development of design tools is facing a major challenge in the better integration of urban analytics, optimization, and generative methods in urban design. Thus, tools are required that can support the different phases of the Urban design & Planning processes and provide an interactive environment to address the various problems and context of urban scenarios. (Koenig, et al., 2020)

2.2 Background

Research into design optimization in computer-aided architectural design (CAAD) has evolved over the past few decades to incorporate not only architectural design, but also urban design. The KaisersRot project is a clear example of the widespread use of CAAD in the practice of architecture and urban planning (Braach, 2014). Much research has been performed on investigating configuration combinations and hybrid optimization approaches in the sense of architectural design. An evolutionary system for architecture discovery was suggested by Janssen (2009). For the architecture exploration of performance-driven geometries, Turrin et al. (2011) developed a system that integrates parametric modeling with genetic algorithms. Methods for integrating generative architecture with evolutionary discovery were also suggested by Stouffs and Rafiq (2015). In the architecture sense, Wortmann and Nannicini (2016) implemented a model-based optimization technique that dramatically enhances the velocity of the optimization process by estimating the fitness landscape, thus reducing the need for each variant to run computationally costly evaluation algorithms. All the approaches suggested so far, though, are typically for single-objective optimization. Hybrid techniques that integrate metaheuristics with other optimization strategies, such as machine learning, have been shown to be useful for architectural architecture optimization problems (Wortmann et al., 2015).

With the advancement of spatial analysis tools, space optimization approaches in the urban environment first started to become the focus of research. A strategy for generating urban patterns through the combination of form grammar and genetic algorithms was proposed by Celani et al. (2011). EMO approaches for land-use planning have been used by Cao et al. (2011), while Motieyan and Saaid Mesgari (2018) suggest an agent-based modeling approach to land-use and transport planning. A synthesis approach for street networks and building layouts based on EMO was introduced by Koenig et al. (2013). "A paradigm called Cognitive Design Computing (CoDeC), which enhances the role of "the person in the loop" with personalized EMO algorithms, was also proposed by Koenig et al. (2018). The explanation for selecting urban design EMO algorithms is that they are well suited for the following situations: I during an urban design process, the problem description varies. In addition, (ii) involving humans in the optimization process will direct the search process and increase the efficiency of solutions, especially with regard to solving design problems where the design space is large, and not all criteria and constraints can be defined in advance (Scott et al., 2002).

2.3 Generative Design

Design problems are both multi-dimensional and highly interactive (Lawson, 2005). They consist of many parameters and goals like economic value, function, appearance, etc., that need to be optimized simultaneously. The traditional approach to design where a human designer creates specific design options based on the experience and intuition limits the exploration of the entire design space.

With the advent of Artificial Intelligence (AI), the computer's power combined with the human designer can help in wide exploration of the design space, incorporating multiple parameters and goals. While the computer has the capability of computing a multitude of parameters and goals and produce multiple design options, the human designer has the experience and the right intuition to decide the best design option. (Rohrmann, 2019)

To achieve this synergy, we can comprehend from nature's evolutionary design approach. The evolutionary process of nature was first described by Charles Darwin in his book "On the Origin of Species" in 1859. The evolution method operates at the species level, which conceals its individual members' unique abilities and properties. All the species' members are uncommon, share common characteristics, and reproduce and create new members.

Over time, the reproductive process continuously improves the species' abilities and characteristics through adaptation and interaction with other species and its environment. This process is called "natural selection." There are three steps involved in natural selection:

1. Selection – The species' members compete for the limited resources, and only those who are most adapted to their environment survive.
2. Breeding – The survivors reproduce and create new offspring that share their characteristics.
3. Mutation – Some of the offspring's characteristics are randomly changed, responding to the natural conditions.

This process ensures enhanced abilities and characteristics of the species' individuals over time and is also commonly known as "survival of the fittest." (Nagy D. , Learning from nature, 2017a)

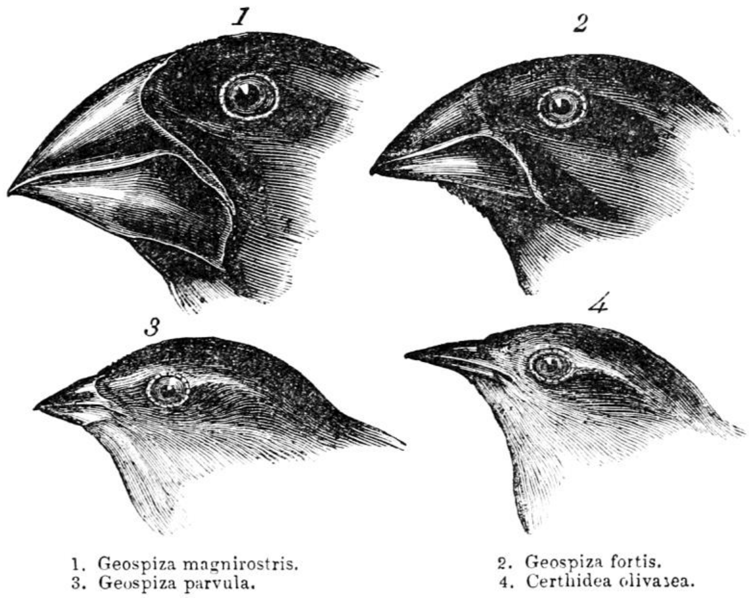


Figure 1: The evolutionary process in nature (Darwin's finches or Galapagos finches. Darwin, 1845)

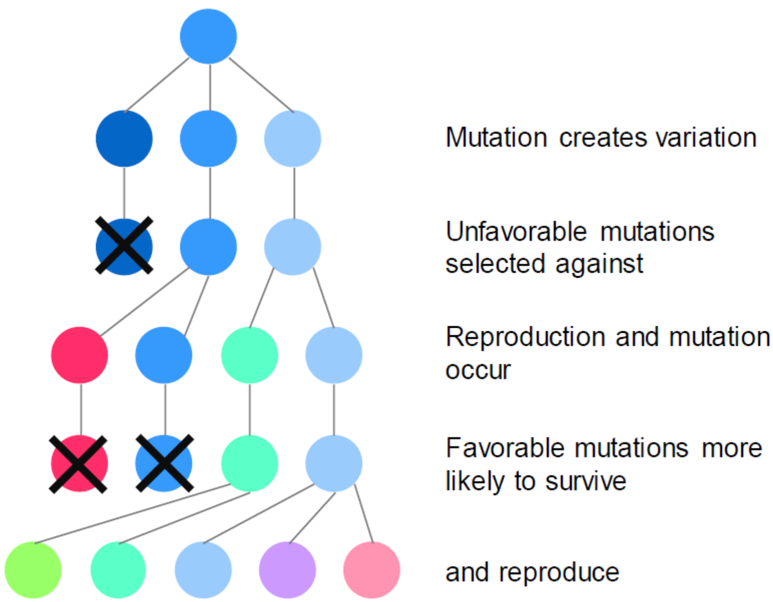


Figure 2: The process of evolution in the nature. (Nagy, 2017)

Generative Design is based on the evolutionary process of the nature. It can generate population of multiple design options, analyze these options, and generate even better performing solutions. The Generative Design falls under the group of “search algorithms” or “metaheuristics.” “Genetic Algorithm” is a search algorithm used for the Generative Design studies, which is a part of evolutionary algorithm. (Nagy, 2017) It is based on the following consecutive principles.

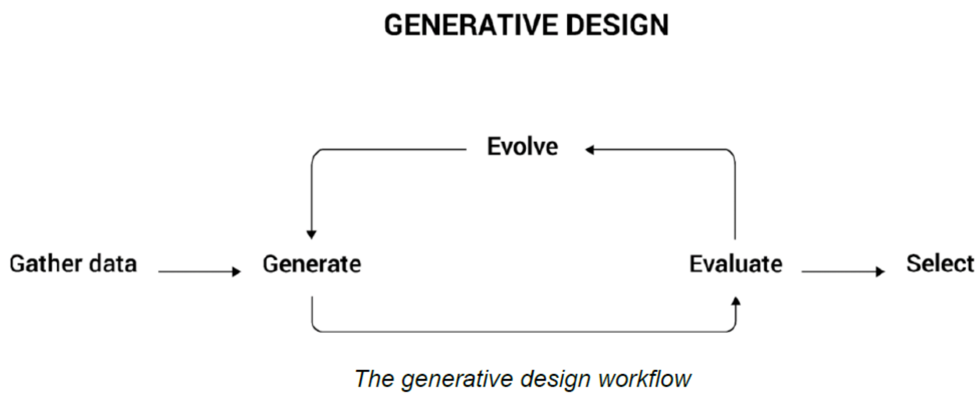


Figure 3: The Generative Design workflow (Walmsley & Villaggi, 2018)

Generate

The algorithm generates all the possible design solutions in the delineated “design space’.

Evaluate

Performance of each design is evaluated.

Evolve

The Evolutionary algorithm finds the unique high-performing design options throughout the design space.

The process of “Generate – Evaluate – Evolve” is looped until a stop criterion is satisfied or an optimum solution is identified by the evolutionary algorithm. The algorithm needs to explore the entire design space to optimize the evolution of the design from one generation to another and eventually identify the desired solution. There are a variety of possible combinations of “genes” in the solution space which are known as the parameters for a parametric model. The blend of these parameter values produces specific outcomes that are further evaluated in each generation. To generate specific outcomes, specific parameters are combined to achieve the desired design goals.

Let us take an example of a glass curtain wall façade where the major design parameters are the number of horizontal and vertical beams. For a constant façade area, the size of the individual panel is a tradeoff between these two parameters. The higher the number of beams, the higher number of glass panels, and vice versa. (Rohrmann, 2019)

Deciding the number of glass panels is a crucial decision for the project manager. As the number of glass panels increase the time of construction also increases. Also, the production of bigger glass panels is hard and are costly. Hence, there is a need to evaluate different possible scenarios, to identify the best suited number of glass panels. So, the major goal of optimization is to reduce the construction time and cost by finding the best suitable number of glass panels.

The solution space for this optimization needs to identify a variety of designs, starting from a single panel (H = 0, V = 0), to a very fragmented arrangement (H = 100, V= 100).

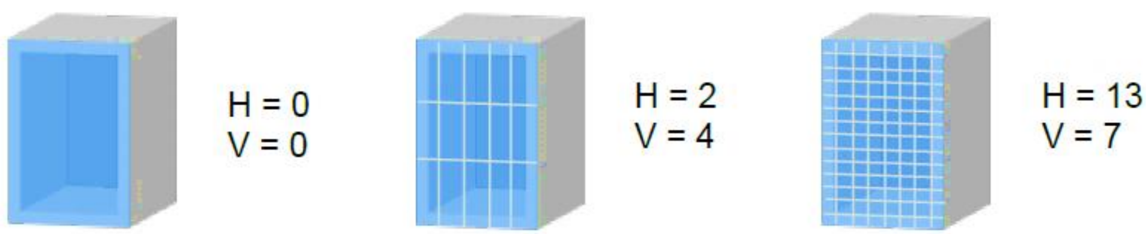


Figure 4: Three different configurations for the façade (Rohrmann, 2019)

The design goals – minimum time and cost of construction of the glass façade, determines the direction of the optimization which is done with the changes in the number of horizontal and vertical divisions. These horizontal and vertical divisions are the parameters that guide the optimization.

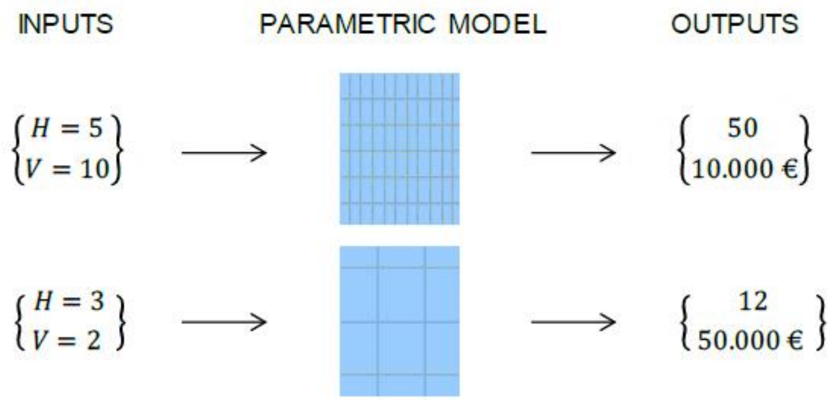


Figure 5: The function of a parametric model. (Rohrmann, 2019)

Hence, these parameters are particularly important for the generative design study. The algorithm continuously changes the values of these parameters to explore the different solutions. Based on the design goals, optimum solutions are selected. (Rohrmann, 2019)

2.4 Multi Objective Optimization

Design problems consist of multiple criteria that need to be optimized at the same time and these criteria either corroborate or contradict each other. From the earlier example, maximizing the number of glass panels reduces the cost but at the same time minimizing the number of panels reduces the construction time. Here, the two objectives, reducing construction cost and time, contradict each other, and an optimization is required to find the best possible outcome.

In the case of two or more competing objectives, finding optimal choices is called Multi-Objective Optimization (MOO). "No single solution exists for a nontrivial multi-objective optimization issue that optimizes each objective at the same time" (Multi-objective optimization, 2019). There is a trade-off condition where no objective function can be increased in value without decreasing the value of other objectives. Therefore, to achieve these targets, there should not be one particular solution, but instead, there are multiple optimal solutions from Pareto. When the objective values cannot be increased without eliminating other values, a solution is considered Pareto Optimal or Non-dominated. Therefore, there is no solution that does better on both targets.

From a mathematical perspective, a Multi-Objective Optimization problem can be described as:

$$min\ (f_1(x), f_2(x), \ldots.\ f_k(x));$$

$$s,t,x \in X;$$

(Miettinen, 1998)

The f1 and fk are objective functions, with k representing the objective numbers. The vector of *X* describes all of the designs that are limited by constraint functions. It is possible to optimize a given objective function by decreasing the negative. “An element $x^* \in X$ is called a feasible solution or a decision that is feasible. Vector $z^* := f(x^*) \in R^k$ for a feasible solution x^* is called an objective vector or an outcome”. (Multi Objective Optimization, 2021)

The Multi-Objective Optimization problem finds the range of nondominated solutions.
A solution $x^1 \in X$ dominates another solution, if

1. $f_i(x^1) \leq f_i(x^2)$ for all indices $i \in \{1,2,\ldots,k\}$ and
2. $f_j(x^1) < f_j(x^2)$ for all indices $j \in \{1,2,\ldots,k\}$. (Miettinen, 1998)

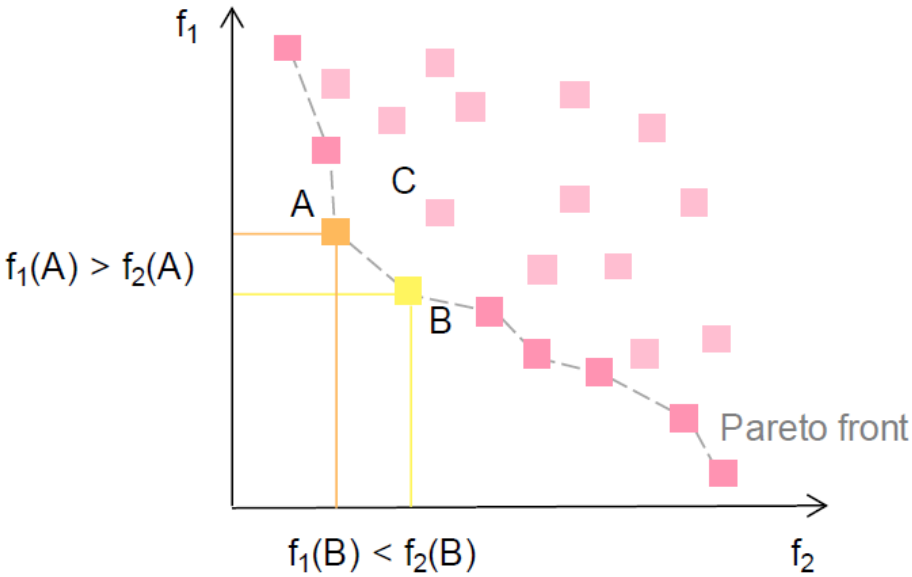


Figure 6: The Graph represents the Pareto frontier formed by solutions A and B. The solution C does not lie on Pareto frontier being dominated by A and B.

All the solutions that are not dominated by other solutions form the Pareto Frontier. The aim of GD is to have a diverse range of solutions that converge along the boundary of Pareto Frontier. This encourages a human user in recognizing that it cannot be more refined, to choose a favorable design approach.

2.5 Genetic Algorithms

Genetic Algorithm is a metaheuristic used to solve multi-objective optimization problems. John Holland proposed the first Genetic algorithm in 1975, in his book “Adaptation in Natural and Artificial Systems”. The ideas of evolution were translated by Holland to form a computational algorithm. His algorithm proved to be much more subtle than a random search algorithm for the space of chromosomes, with the preservation of the best. (Holland, 1975)

Workflow of a Genetic Algorithm is as follows:

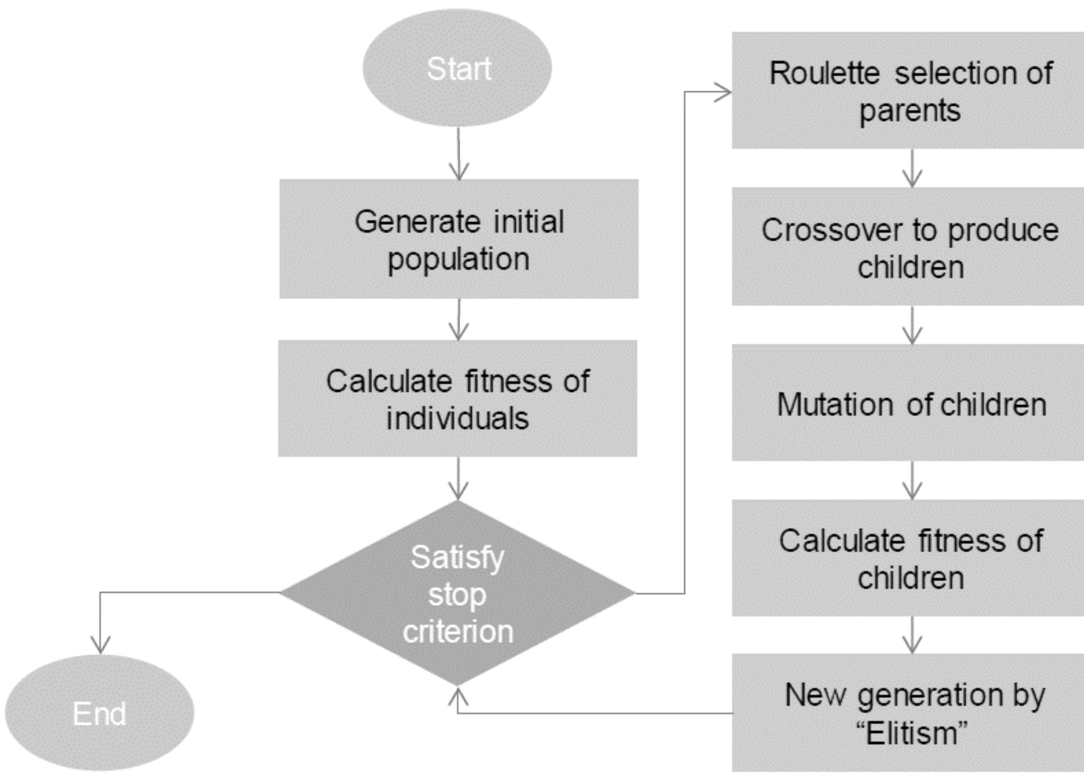


Figure 7: Basic workflow of a Genetic Algorithm (GA) (Nagy D. , Learning from nature, 2017a)

The first generation of individuals is generated that form the parent population. First generation fitness is calculated, if one of solutions (individuals) satisfies the fitness criterion, the algorithm ends. If none of the individuals satisfies the stop criterion, the evolutionary principles of selection, crossover and mutation are carried out. Two of the first-generation individuals are chosen to mate, their chromosomes are crossed together to create a new entity, the first-generation infant population. Then, to put in random new, probably fitter DNA, mutation is added randomly to the children.

Children's population fitness is then measured. Both first-generation individuals are sorted by their reached fitness level, along with the parent population. The method of rating the population of children along with the population of parents is called elitism. It guarantees the retention of good solutions discovered previously. In the optimization process, the highest-ranking individuals form the next generation.

Since the evolutionary model developed by Holland, different variants of Genetic Algorithms have been proposed. For the new parent population, they differ more in the way they rate and pick solutions.

Early Genetic Algorithms (GA) with low complexity of computation but no elitism are:

- NPGA (Niched Pareto GA)
- MOGA (Multi-objective GA)
- VEGA (Vector evaluated GA)

(Hong, 2012)

These Genetic algorithms fail to provide a wide variety of solutions and are very slow to converge to the Pareto frontier. This makes them undesirable to use for complex computations. However, elitism provides good solutions and speed up the process of GA (Agarwal & Et. Al. , 2002). The three commonly known elitist Genetic Algorithms are:

- PAES – Pareto Archived Evolution Strategy
- SPEA – Strength Pareto Evolutionary Algorithm
- NSGA – Nondominated Sorting GA

2.6 Non-Dominated Sorting Genetic Algorithm II (NSGA – II)

The NSGA Genetic Algorithm uses non-dominated sorting to compute the crowding distance and dominance rank to identify a diverse set of solutions. Earlier, the NSGA algorithm lacked elitism and had a computational complexity of $O(MN^3)$. M represents the objectives and N represents the population. The cubic population size made the algorithm slow. In 2000, NSGA-II was introduced, which included the elitist approach and has a computational complexity of $O(MN^2)$.

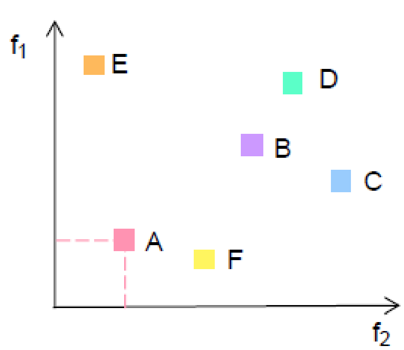
To solve a Multi Objective Optimization problem, an algorithm needs to accommodate the following goals:

- Its conservation of previously detected non-dominated points
- Its rate of movement towards the front of the Pareto
- The range of points that it offers on the Pareto front
- Its capacity to offer a sufficiently sized number of solution points for preference to the human decision-maker.

The workflow of the NSGA – II algorithm is as follows:

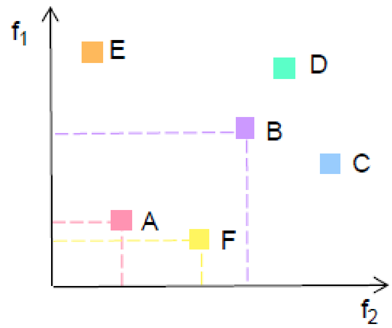
2.6.1 Fast Nondominated Sorting

A short method of sorting solutions into groups is used by the NSGA-II. Each solution p is provided with a set of solutions s_p and domination count n_p. The n_p represents the count of solutions that dominate the selected solution and s_p represents the group of solutions that it dominates. (Agarwal et. al. , 2002)



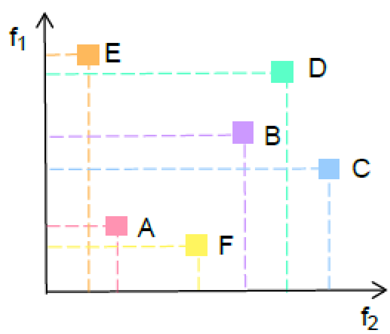
	n_p	S_p
A	0	
B		
C		
D		
E		
F		

Figure 8: Here, no solution dominates A; n_p = 0.



	n_p	S_p
A	0	{B}
B	2	
C		
D		
E		
F		{B}

Figure 9: The domination count of B is 2 as it is dominated by A and F.



	n_p	S_p
A	0	{B, C, D}
B	2	{D}
C	2	null
D	3	null
E	0	null
F	0	{B, C, D}

Figure 10: Here, all the domination scores and sets are represented, for each solution.

fast-non-dominated-sort(P)
for each $p \in P$
  $S_p = \emptyset$
  $n_p = 0$
  for each $q \in P$
    if $(p \prec q)$ then — If p dominates q
      $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
    else if $(q \prec p)$ then
      $n_p = n_p + 1$ — Increment the domination counter of p
  if $n_p = 0$ then — p belongs to the first front
    $p_{\text{rank}} = 1$
    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$
$i = 1$ — Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$
  $Q = \emptyset$ — Used to store the members of the next front
  for each $p \in \mathcal{F}_i$
    for each $q \in S_p$
      $n_q = n_q - 1$
      if $n_q = 0$ then — q belongs to the next front
        $q_{\text{rank}} = i + 1$
        $Q = Q \cup \{q\}$
  $i = i + 1$
  $\mathcal{F}_i = Q$

Figure 11: Code for NSGA-II Algorithm. (Agarwal & Et. Al. , 2002)

The solutions with $n_p = 0$ lie on the first pareto frontier. Now, for the solutions with domination score = 0, the domination count is reduced by one for the members of its set.

	n_p	S_p
A	0	{B, C, D}
B	2	{D}
C	2	null
D	3	null
E	0	null
F	0	{B, C, D}

→

	n_p	S_p
A	-	-
B	0	{D}
C	0	null
D	1	null
E	-	-
F	-	-

Figure 12: Domination score reduced to find the next Pareto Frontier group.

	n_p	S_p
A	-	-
B	0	{D}
C	0	null
D	1	null
E	-	-
F	-	-

→

	n_p	S_p
A	-	-
B	-	-
C	-	-
D	0	null
E	-	-
F	-	-

Figure 13: Domination score again reduced by 1 to group the third pareto frontier.

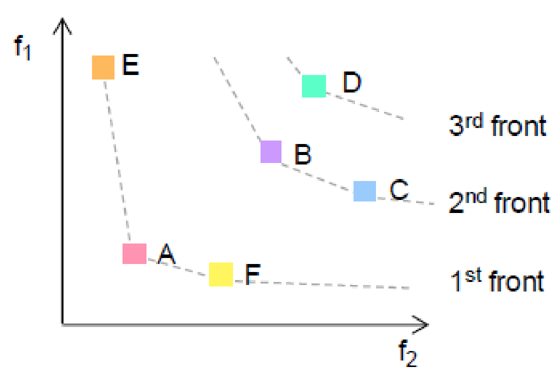


Figure 14: Final Sorting of the Solutions.

The process of reducing the Domination ranks continues, until all the solutions are arranged in different fronts. (Agarwal & Et. Al. , 2002) (Rohrmann, 2019)

2.6.2 Crowding Distance Computation

As previously stated, it is needed that the algorithm also preserves a degree of variation in its solutions along with convergence to the optimal Pareto front. This helps discourage local minimum alignment and provides the human decision-maker with a range of alternatives. This means that a solution from a lower populated area is preferred to an individual solution in a crowding region because it is at the same exercise stage. The NSGA-II set up the crowding distance equation to measure the density of the neighborhood of a solution.

On the either side of the solution p, average distance is calculated along each objective. (Agarwal & Et. Al. , 2002)

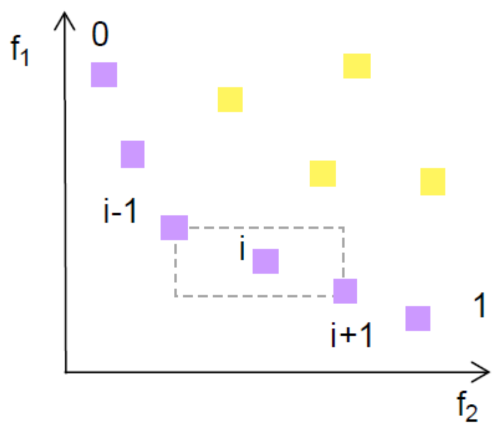


Figure 15: Calculation for Crowding Distance.

The crowding distance for the i^{th} solution is calculated by forming a rectangle adjoining the two closed nearby solution points. For every solution, a crowding distance ($i_{distance}$) and a nondominated front rank i_{rank} is provided by the algorithm. Now, the crowded comparison operator is used to compare two solutions. The operator prefers solutions from less crowded region. (Agarwal & Et. Al. , 2002)

2.6.3 Main loop

For every solution generation (t), the parent population can be represented as P_t and the offspring population as Q_t. The combined population is $R_t = P_t \cup Q_t$.

The combined population R_t is sorted, as described in the last chapter, to form the pareto fronts. Now, from the top solutions of the previous generation, the next generation, P_{t+1} should be formed. If the size of F1 is less than N, P_{t+1} is passed to all members of F1 until P_{t+1} contains N members, subsequent sets are picked in the same way. To find the best solution to complete the population, crowding distance sorting is extended to the set if the last selected set has more members than the remaining positions in P_{t+1} (Agarwal & Et. Al. , 2002).

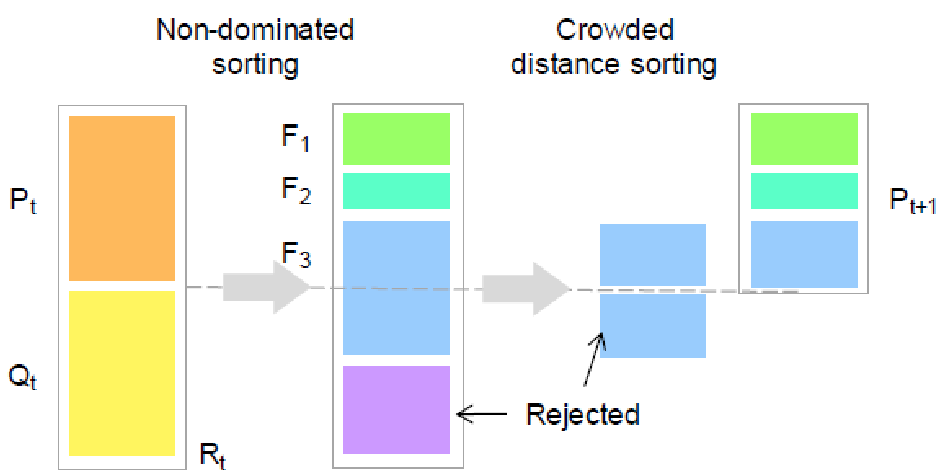


Figure 16: NSGA-II Workflow

In the main loop, sorting based on rank is not required, making the algorithm faster. Through selection, crossover and mutation, new offspring population Q_{t+1} is created from the parent P_{t+1}.

2.6.4 Tournament selection

The parent population members are selected to generate offspring. After the mating happens between the members of the parent population, several solutions are generated. The solutions that are better performing are selected to procreate again so that better solutions are generated that have high fitness. Sometimes, when there is a high pressure of selection of better performing solutions or there are many solutions that make up to the pareto front, tournament selection known as “binary tournament selection” happens.

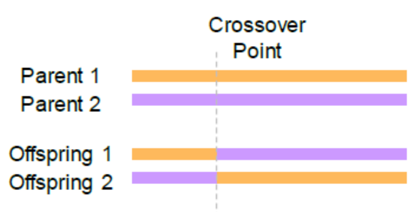
Here, tournament is conducted between the individual solutions, to identify a higher performing individual solution. This is done based o the crowding distance of the solutions. The individuals with a lower i_{rank} wins. The tournament is repeated until the sorting is done or optimum solution is achieved. (Agarwal & Et. Al. , 2002)

2.6.5 Crossover and Mutation

Crossover method combines the chromosomes or the properties of two parent individuals to form new offspring. Chromosome is defined as the design variables of a parametric model. For example, for the Glass Façade, the Horizontal and Vertical divisions are chromosomes. The chromosomes of the parent solutions are combined to form new offspring.

The various ways in which chromosomes of parent individuals is combined are as follows:

a. Single Point Crossover: Two parents are selected randomly, the chromosomes of the two parents are combined at a crossover point. Both the offspring inherit the properties of the parent individuals.



b. Two Point or Multiple Point Crossover: The same process is carried out for multiple point crossover, the properties are divided at multiple points, among the offspring. This ensures multiple offspring.
c. Uniform Crossover: A binary mask is generated randomly in the Uniform crossover method. This Mask has the length same as the count of Parent chromosomes. The crossover mask becomes the basis for the crossover of properties among the offspring. If the value of the mask is 1, the property is taken from the first parent. If the value is 0, properties are taken from the second parent. This process is flipped for the second child.

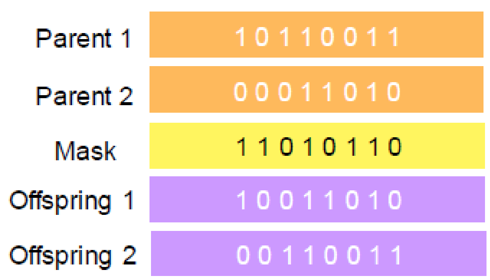


Figure 18: Example of a uniform crossover.

The other crossover methods include – Shuffle crossover, partially matched crossover, 3 – Parent crossover, etc.

The crossover among the parent individuals ensure that the offspring attain the good properties of the parent individual, to ensure the optimization of new solutions. The crossover process do not add any new properties or information to the new offspring and ensure full exploration of the design space. This also ensure that the algorithm avoids the local minima.

Mutation

The mutation method includes addition o a mutation chromosome which dictates the addition of properties of the new offspring. There are three most common types of mutation:

a. Flipping

Based on the Probability of mutation (P_m), the flipping happens between the parent chromosome and the mutation chromosome. The higher probability of mutation, higher number of bits are flipped.

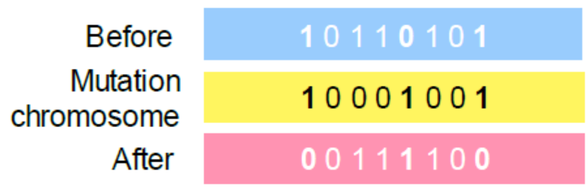


Figure 19: Flipping Method for Mutation

b. Interchanging

In this method, two positions of the parent chromosome are selected randomly, and their bits are interchanged.

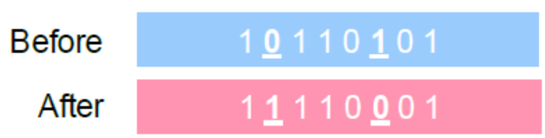


Figure 20: Interchanging Method for Mutation.

c. Reversing

Two adjoining bits of the chromosome are selected and switched.

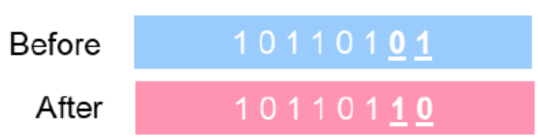


Figure 21: Reversing Method of Mutation.

2.6.6 Summary

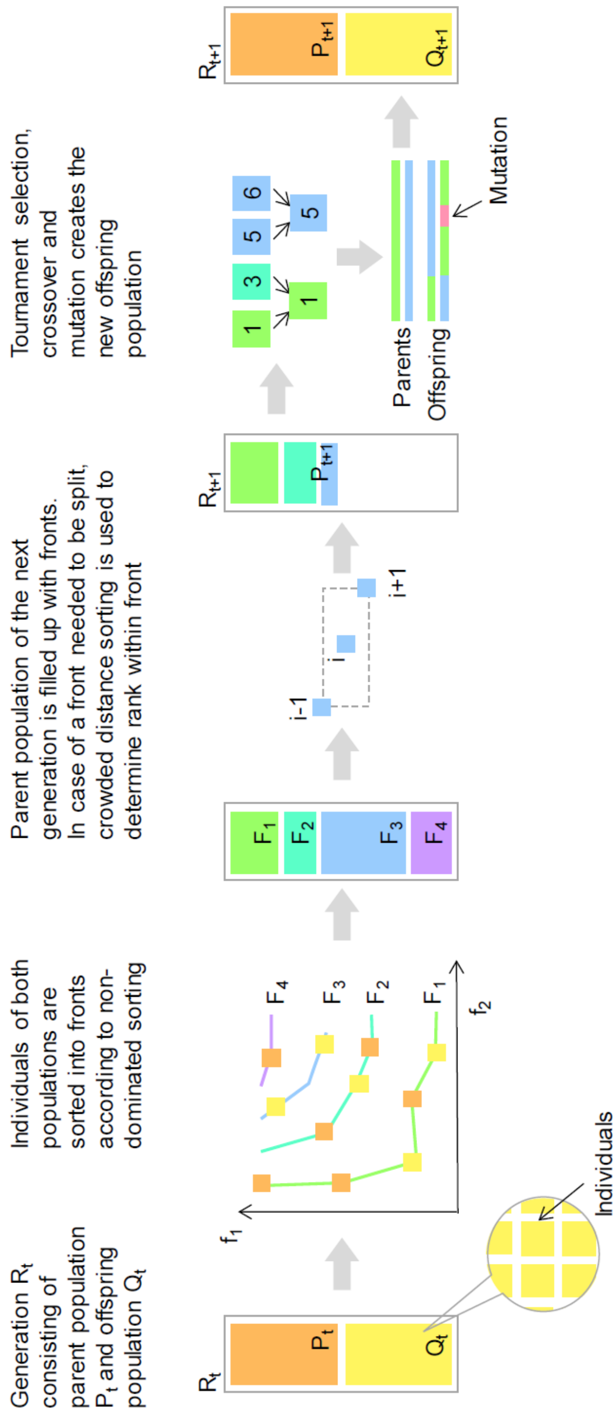


Figure 22: Workflow of a NSGA-II Algorithm. (Agarwal & Et. Al. , 2002)

2.7 Case Study

2.7.1 Case Study 1 – Computational Urban Design Prototyping: Interactive Planning Synthesis Methods – A case study in Cape Town.

- (Koenig, Reinhard, et.al., 2018)

2.7.1.1 Project Details

Project Type – Urban Neighborhood (Research Project)

Research Partners:

a) Future Cities Laboratory, Singapore
b) Department of Energy, Austrian Institute of Technology, Vienna, Austria
c) Departement Architektur, Eidgenössische Technische Hochschule Zürich, Zürich, Switzerland

Authors – Yufan Miao, Reinhard Koenig, Katja Knecht, Kateryna Konieva, Peter Buš and Mei-Chih Chang.

Project Location – Cape Town, South Africa

Year of Research – 2018

Tools Used:

a) Rhino 3D for Modelling
b) Grasshopper for Visual Scripting
c) Empower Shack Project for Live interaction with the Model on Browser.

2.7.1.2 Introduction

The traditional methods of Urban Design follow sectoral and static approaches in a time when the use of computers have become so prominent in the different sectors of design processes. Automation using computational tools, in the context of Urban Design helps the designers, by providing a comprehensive overview of the multiple design options for a complex set of requirements in faster time duration. This method is called CUDP – Computational Urban Design Prototyping. (Koenig, et.al., 2018)

The Computational Urban Design Prototyping Project is carried out for the city of Cape Town, South Africa. The city accommodates around 7.5 million people living in informal settlements. The city has a need of 2.5 million housing units. The research aims to develop new design and analysis tools for a comprehensive urban spatial design that can be implemented on different sites. The tools developed in the project is adaptable to meet site specific requirements and needs of different stakeholders.

The CUDP project developed in the Research study has two major characteristics:

a. The project allows designers to create multiple design options by adjusting parameter values.
b. Application allows the designers to change the geometric constrains of the project, making it more interactive.

The Project has been developed using the software Rhino 3D with Grasshopper add-on. Parameters of the study have been designed based on the requirements of the local stakeholders.

2.7.1.3 Aims of the Project

- Development of Data structure for Spatial configuration of Streets, Blocks and Parcels.
- Translation of stakeholder requirements into parameters for the Computational Algorithm.
- Generation of spatial configuration for: Optimized use of land for dense housing configuration; Efficient space allocation; Neighborhood preferences of the community.
- Maximize the spatial qualities, level of details and geometric precision of the automated design.

2.7.1.4 Methods Adopted

The following section represents the data structure used for the generation of spatial configuration comprising of the street networks, land parcels and buildings. The authors have developed a CPlan Library, which is a Software Package for Computational Urban Design Prototyping, using Forge. The CPlan opens source library is capable of computational urban analysis and planning. The software is written in C# and C++ languages.

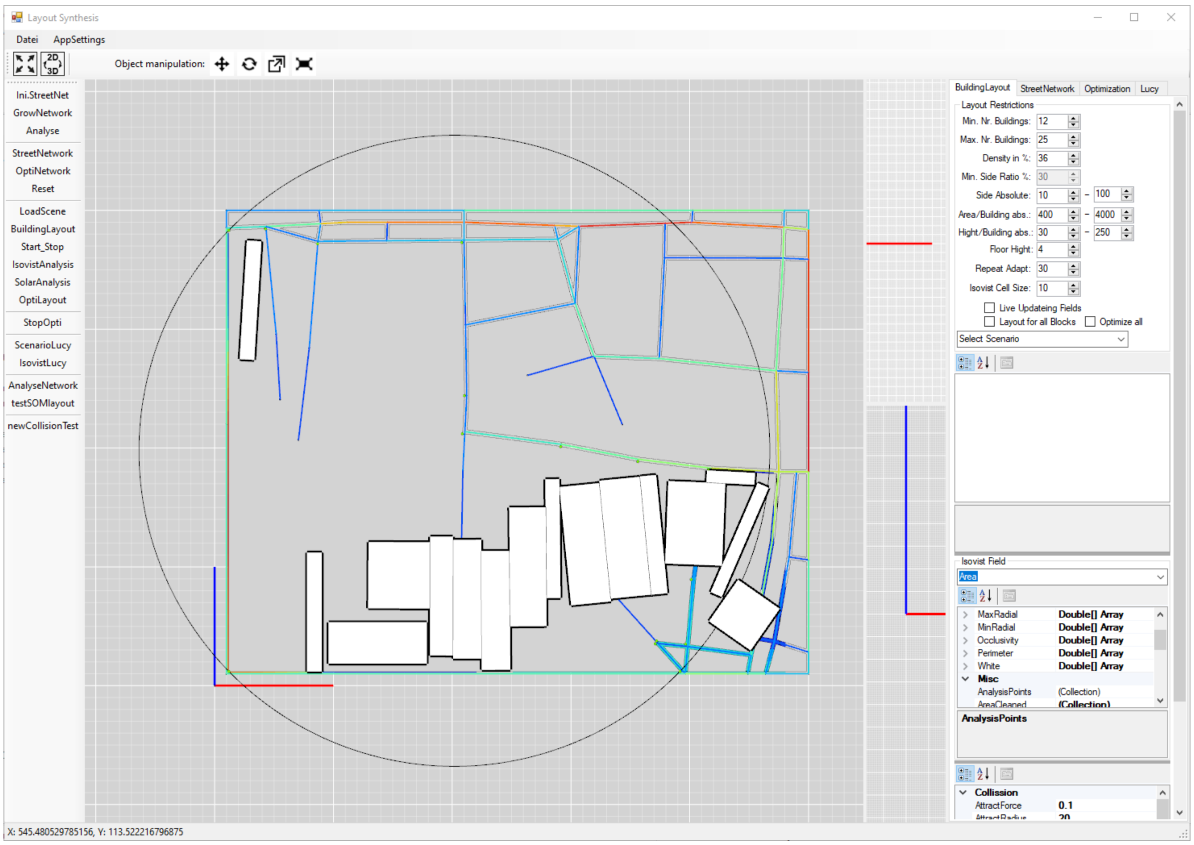


Figure 23: CPlan Software Interface

Data Structure followed in the CPlan Framework for the generation of the Spatial configuration is as follows:

a. Street Networks represented as instruction trees.

The Figure 24 represents the instruction tree for the generation of street networks. Nodes are used to contain the information for addition of street segments. Each node contains the information of length, angle, and degree of connectivity. The edges between the nodes define the relationship of two nodes and the parent node.

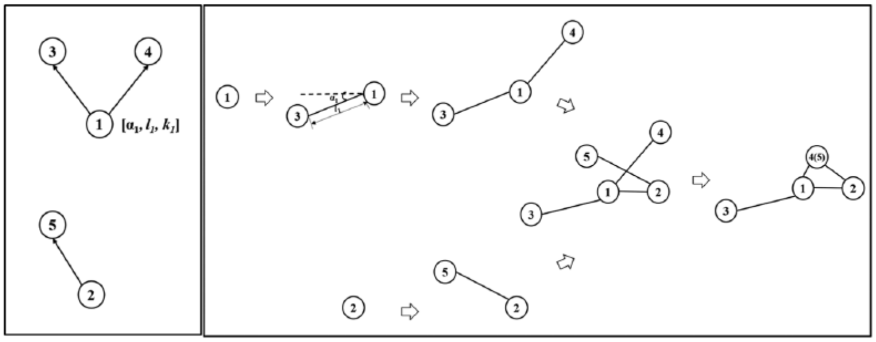


Figure 24: Street Network Instruction Tree

b. Slicing Tree representation of Parcels.

Figure 25 represents the slicing tree structure for the parcellation of land. The V represents the Vertical slicing and H as Horizontal slicing. The index of the slicing tree shown below is represented in a tree like structure. At the bottom, land parcels are represented which emanate from the slicing lines represented as streets.

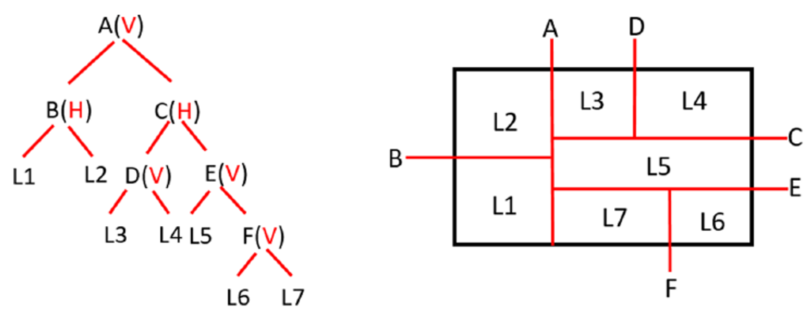


Figure 25: Land Parcel representation as slicing tree.

Figure 26 represents the method used by the slicing tree from top to down, the step wise parcellation of land is represented. The Node A divides the site vertically and emanates two slicing nodes (B and C) horizontally dividing the land. Similarly, the process is followed to get the land parcel geometry.

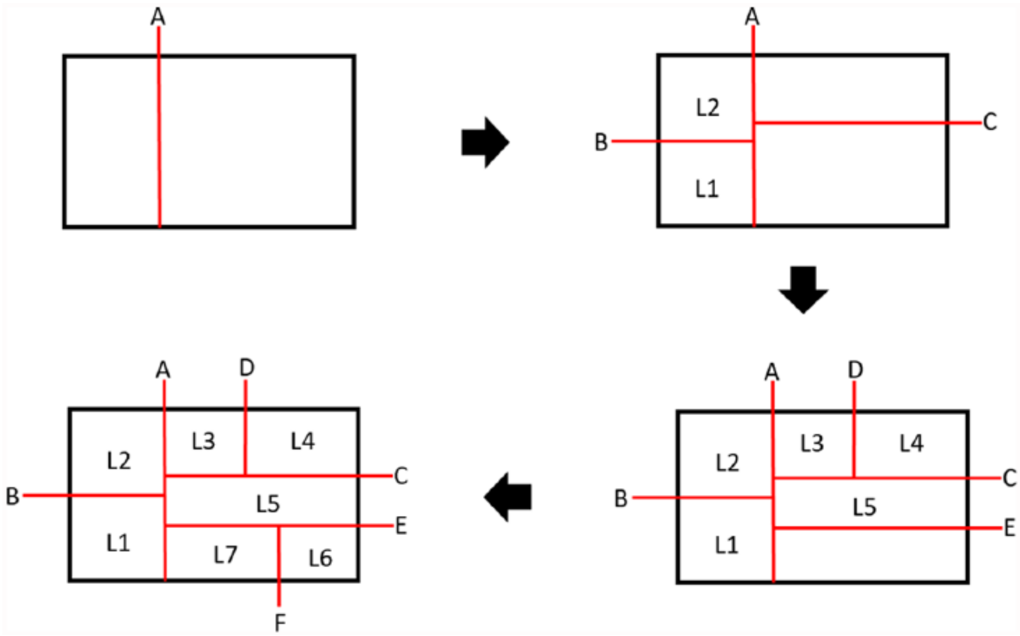


Figure 26: Method of land parcellation using slicing tree.

When the site has an irregular geometry, bounding box is used which encloses the site, connecting from the smallest edge of the site geometry. In figure 27, the blue triangle represents the site geometry, and the red rectangle is the bounding box. The code identifies the smallest edge and aligns the bounding box to enclose the entire site.

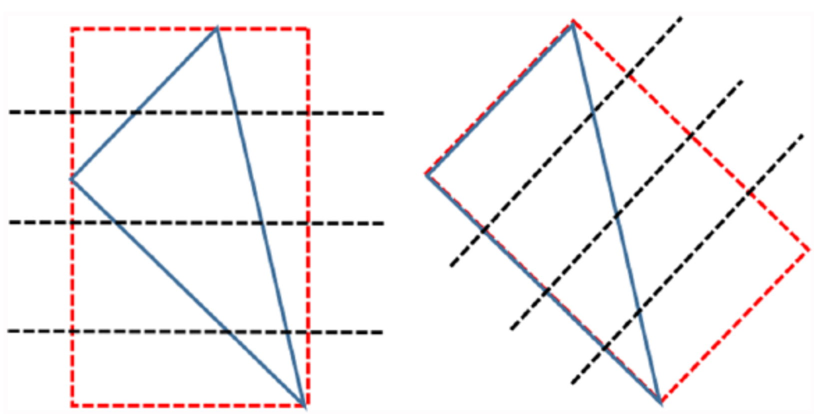


Figure 27:Bounding Box for irregular site geometry.

Additionally, the slicing tree data structure can be designed to include the edge data of the slicing line. This enables to include slicing for uneven shapes. Instead of including the Horizontal (H) or Vertical (V) data, the slicing tree data structure store the data to the edge which it is slicing, along with the length, connection, and angle of the slicing. For example, in Figure 28, the data tree stores edge data for slicing line.

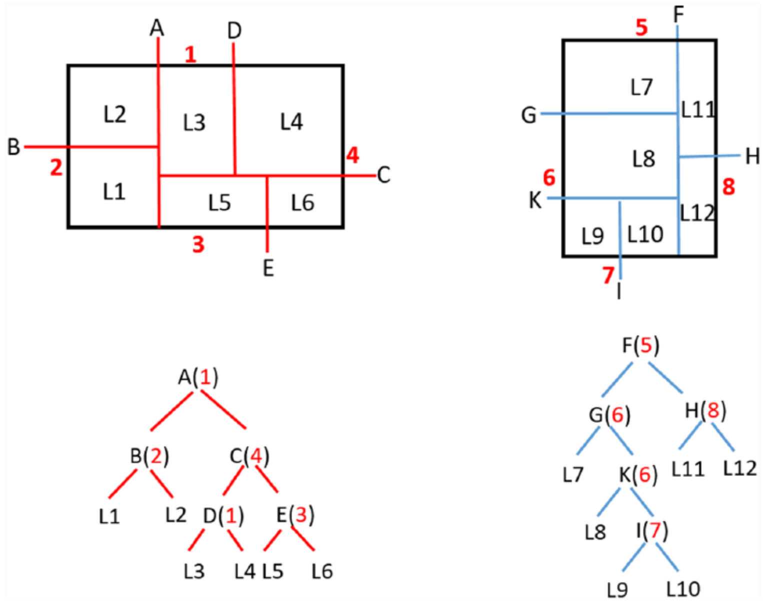


Figure 28: Updated Slicing Tree data structure to include edge identity.

c. Evolutionary Multi Objective Optimization (EMO)

The EMO is applied by recombining the data tree structure through crossover and mutation. The Parent population is crossed over and mutated to create offspring solutions that are better optimized for street networks. The process continues until the goals of the design is achieved. These goals can be of different types, such as: Minimum road length, right angular roads, different parcel configurations, etc.

Figure 29 shows the process of crossover among two parent solutions to create new offspring population.

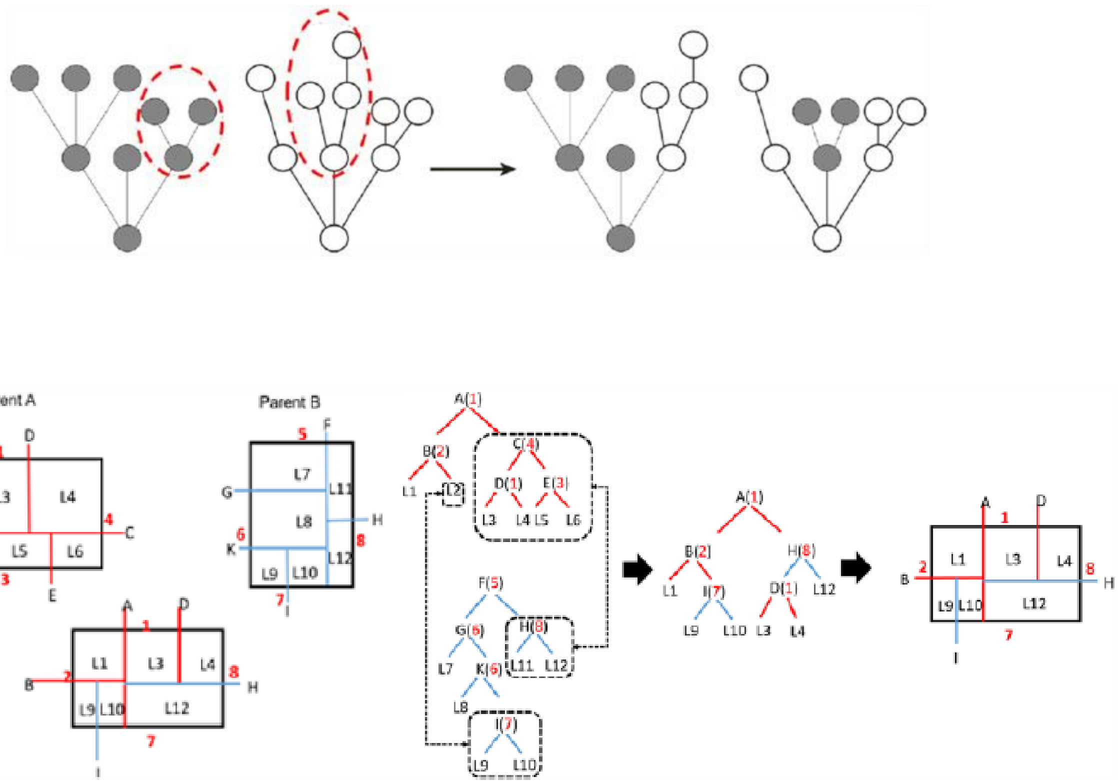


Figure 29: Crossover process for Evolutionary Multi Objective Optimization.

2.7.1.5 Implementation on Site

The project has been implemented in Enkanini, Cape Town, South Africa. There are approximately 7.5 million people living in informal settlements and 2.5 million housing units are required. Hence, there is a need for a fast comprehensive approach that identifies urban layouts faster, optimized and gives the stakeholders, urban planners and designers multiple design iterations that incorporate different goals of the project.

Figure 30: Study Area - Enkanini in Cape Town City, South Africa

The project requirements had been identified together by the local stakeholders, urban planners, and the project team. The major requirements were:

i. Creation of a tool that provides efficient densely packed urban design.
ii. Accessibility to facilities
iii. Fair allocation of Private and Public spaces.
iv. Involvement of the community into the design process through interactive 3D Design.

To address these requirements, the project team developed a computational prototyping algorithm that requires the designer to only specify the site layout, initial street segment and the uniform dimension of the plots (width and depth). The block and parcel generation has been nested to get desired plot sizes. The definition first creates street network based on the specified initial segment, then the blocks are created, where the individual land parcels are divided. For blocks with a depth of 2.5 times that of land parcel, the depth is offset to contain a courtyard and the remaining area is subdivided into plots. Buildings are placed based on the outlines of the individual plots.

Figure 31: Urban Layouts generated by the algorithm. Left: Without any specified street segment. Right: With specified Street segment.

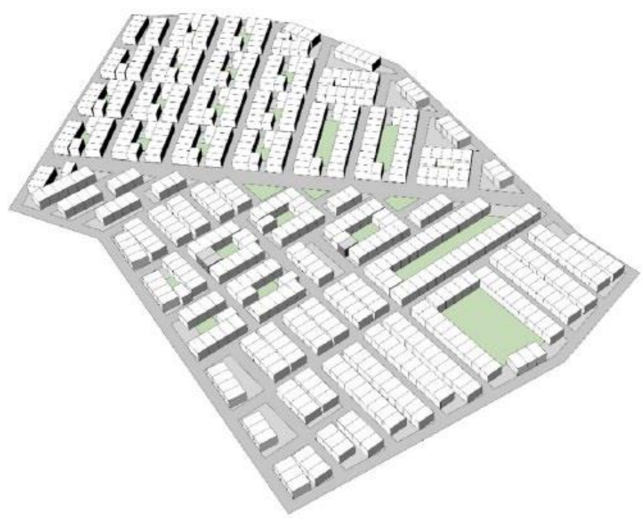


Figure 32: Adding building mass to the generated output.

2.7.1.6 Design Variants

The project team used Speckle plug-in for Grasshopper to make the generated layouts available online for different stakeholders. It also permits the stakeholders to interact with the model by allowing them to change the dimension (width and depth) of the parcels using number sliders. The resulting geometry can update based on the inputs given for the width and depth.

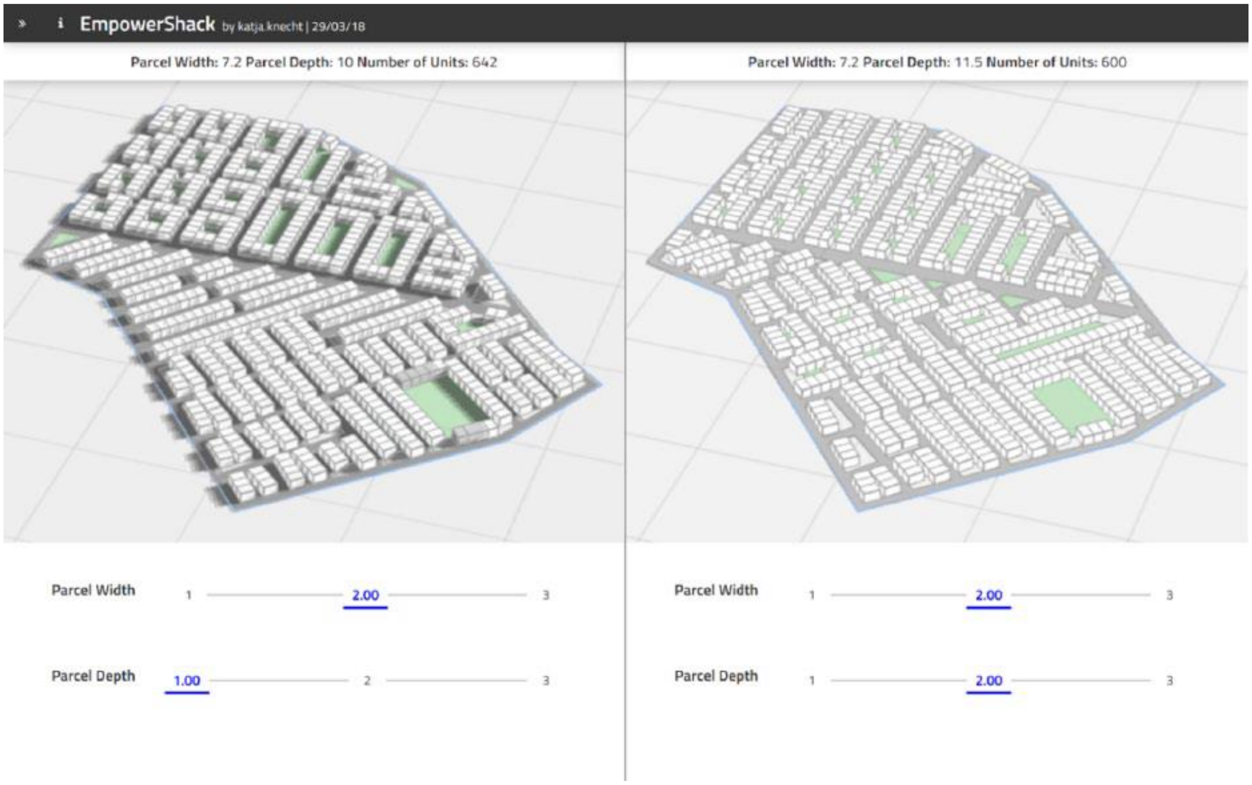


Figure 33: Empower Shack Project for online viewing and interaction with the model.

2.7.1.7 Design Evaluation for Pedestrian Accessibility

The Urban Layouts have also been evaluated for Pedestrian accessibility by gravity-based methods. The distance along the street networks is calculated from each housing unit. The number of accessible facilities is identified is inverse to the travel cost. In this project, only educational facilities have been considered. Figure 34 shows the pedestrian access map. The warm colors in the map represent high access, while cold colors represent low access.

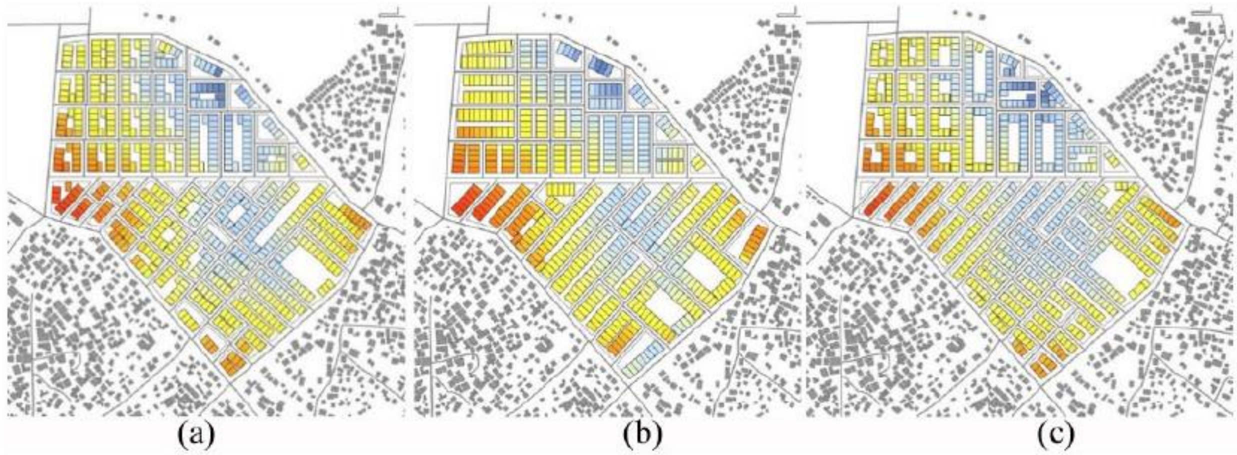


Figure 34: Pedestrian Accessibility Map.

2.7.1.8 Result and Conclusion

The Project has developed a Data Structure for Urban Geometry generation and Multi Objective Optimization to find the optimized solutions. The data structure allocates Street patterns and slices for blocks and plot division, based on the project goals. The preference-based geometry generation incorporates the local stakeholders in the design process. The optimized design solutions are also analyzed for pedestrian accessibility through gravity-based model attractor points, by calculating the distance of the amenities from the individual land parcels.

Methodology adopted in the project:

1. Identification of Project requirement from Urban Planners and Local Stakeholders
2. Creation of Data Structure for the Algorithm
3. Geometry Creation
4. Implementation on Site
5. Analysis of Generated Solutions

2.7.2 Case Study 2 – Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout

(Villaggi & Nagy, 2018)

Through the design of a real-world residential neighborhood development project in Alkmaar, Netherlands, this study demonstrates the application of Generative Design to an urban scale. The case study demonstrates potential urban design complexity by optimizing for two important goals: the developer's profitability (cost and revenue of the development project) and the potential for power generation from solar panels installed on the roofs of the buildings, not only for minimizing the development's environmental impact but also for future homeowners who will benefit from the development. The architectural challenge was to build out a residential neighborhood on an existing 7,000 sqm property. Several restrictions and needs were gathered and divided into two categories: site restrictions and program requirements. Program requirements synthesize the developer's programmatic project goals, whereas site limitations identify those restrictions arising from local building regulations and existing topographical elements. The site constraints and program criteria are listed below:

The site constraints included:

• A specified site border that delineates the Generative Design zone was one of the site constraints.

• Fixed unit orientations that are orthogonal to the site's existing streets

• A maximum building height of 5 floors in the south and 3 floors in the north

• A minimum of one access road to the lot's west and south sides.

• Parking lots must be at least 5 meters away from road crossings.

• Only same house unit types can be adjacent to each other and can aggregate only laterally

The program requirements included:

• There should be at least 4 single houses in the layout.

• There should be at least 3 single houses in the layout

• There should be at least 2 single houses in the layout

• There should be at least 3100 sqm of apartment units in the layout.

A parametric model was constructed based on the limitations and requirements, which can provide a vast range of legitimate design possibilities based on a limited number of input parameters, as seen in the image below:

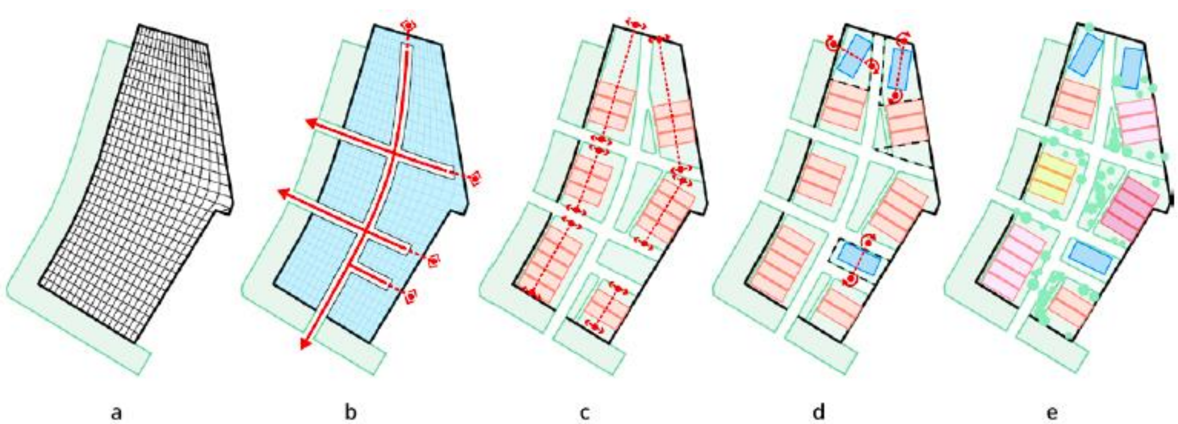


Figure 35: Parametric model description for generating each design option

The programme requirements were represented as an optimization goal, while the site restrictions were immediately included into the model. This enables the discovery of design solutions that, while not completely achieving certain aims, may offer surprising layout strategies that prioritise other goals.

The model is built on a subdivision grid that adapts to the margins of a lot's boundaries (figure 1, a). The grid's margins are used to designate cross-site streets that run in both directions (figure 1, b). The streets split the lot into zones, which are compared to internal model limitations such minimum region aspect ratios and surface areas. Those that don't fit these criteria are either split (creating new streets) or merged (removing the separating street). Green public areas, dwelling units, apartment structures, and pedestrian paths abound in each of the resulting zones (figure 1, c-e). Each road also has parking places along one of its edges. 5 continuous floats with domain [0.0, 1.0] are used to parameterize the model. The difference between the total selling price of the units and the total cost of the project, as estimated by the calculations below, is the project's profitability.

Total selling price = total house price + total apartment price
Total project cost = land cost + construction cost + development cost + selling and rent cost + profit and risk factors

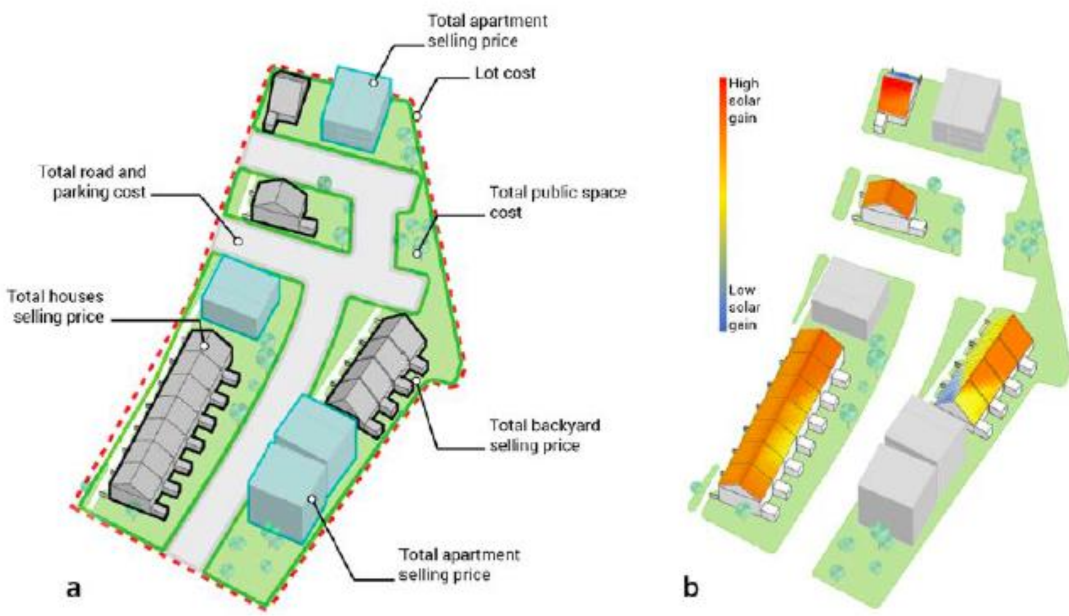


Figure 36: Description of two metrics, development profitability (a) and potential for solar gain (b).

Each roof surface is assessed for occlusions against 48 sun beam vectors to determine the possibility for solar energy gathering (fig. 2, b) (based on 15-minute increments on equinox and solstice dates). The solar energy number that result is the average use of potential solar energy by the roofs of all structures on site, and it is maximized during optimization. They begin to see the relationship between the two aims and the potential trade-off between them by looking at numerous different designs, as indicated in the diagram below:

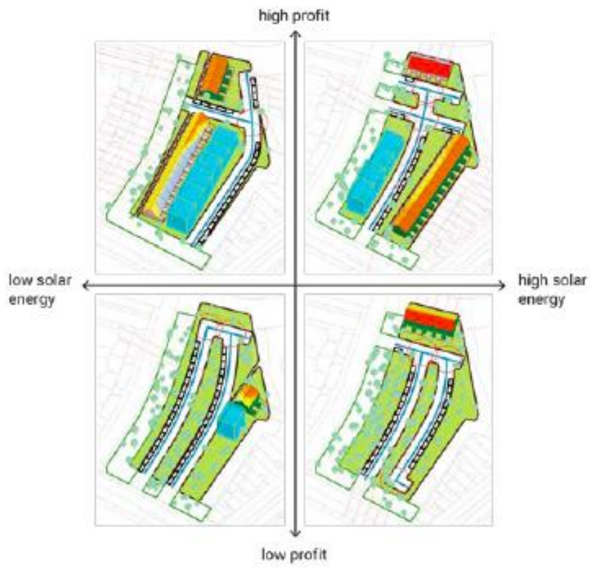


Figure 37: Relationship and trade off goals at performance extreme

This high-level analysis shows potential conflict between two aims, as well as chances to find high-performing designs that maximize the trade-off.

They used a visualization technique to evaluate these features inside a given design space, as demonstrated in the image below for this project (fig. 4). This study was carried out prior to optimization in order to guarantee that the optimization process is productive and produces good outcomes. Because it can create a number of design options, this research reveals that the design space is not too biased. At the same time, it isn't excessively varied because each design reflects a valid solution that adheres to the design problem's limits.

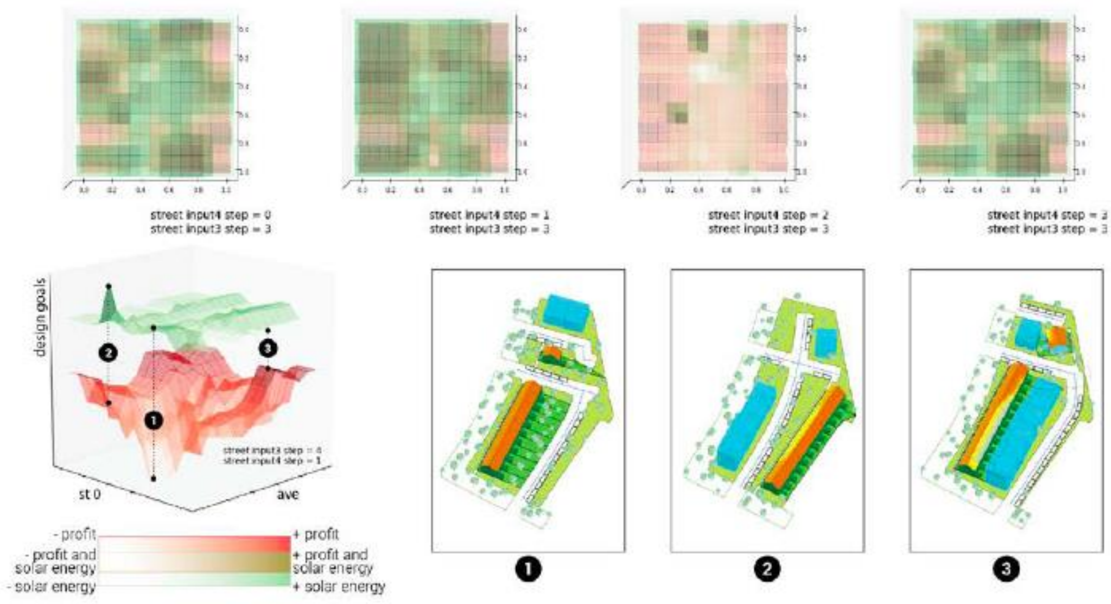


Figure 38: Design space visualization with plot x and y axes representing input parameters and z-axis and color representing averaged values of output metrics

It employs a Genetic Algorithm based on the NSGA-II algorithm to locate designs inside the design space that maximize the values of the two objectives to generate the final design solution. The optimization results are shown in the figure below, with each created design represented as a single dot in the scatter plot, based on the two objectives. The highest-performing designs are those that are closer to the upper right corner, with all of the designs along the pareto boundary representing the optimal trade-offs between the two objectives. The highest-performing designs are those that are closer to the upper right corner, with all of the designs along the pareto boundary representing the optimal trade-offs between the two objectives. At the end, three designs were picked from the entire set to show three distinct strategies for planning out the land (as shown figure below).

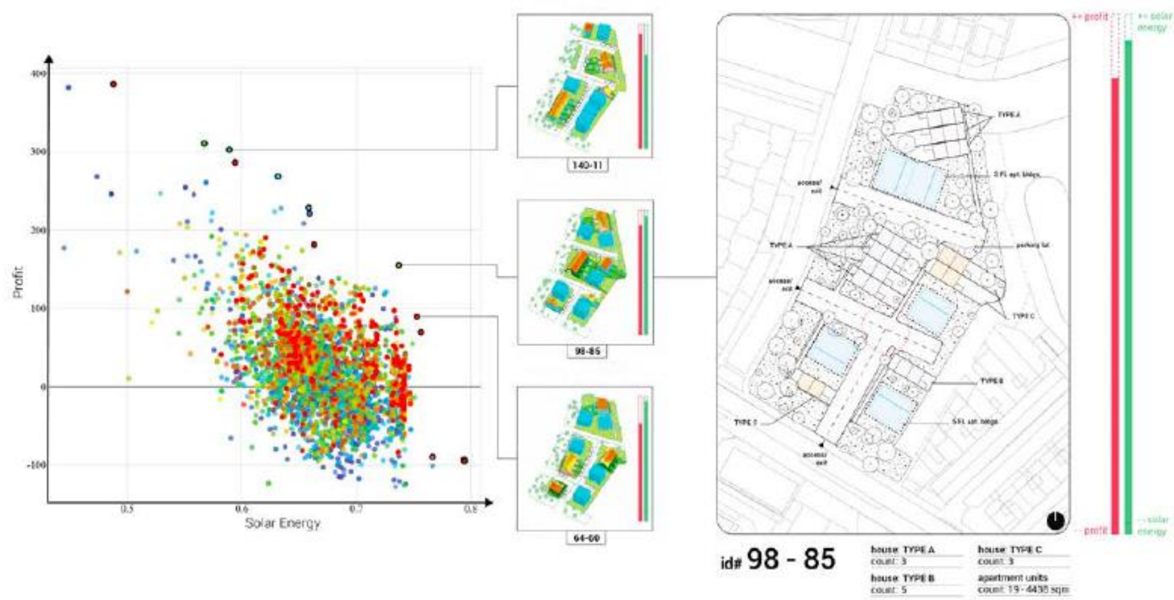


Figure 39: Plot showing trade-off between the two objectives & 3 chosen high performing design

This page is intentionally left blank.

3. Site Analysis

The site for the study has been selected as the CD Block of Salt Lake City, Kolkata. The Generative Algorithm for automated neighborhood layout has been developed based on the existing Block dimensions of the site.

The existing site has been analyzed based on the design goals selected for the generative algorithm.

The analysis includes Site Dimensions, Land Use of the site, Figure ground analysis, Road dimensions, plot dimensions and other spatial characters along with the proximity and shading analysis.

The analysis is done to compare the existing design to the computer-generated design of the same neighborhood.

3.1 Site Introduction – Salt Lake City, Kolkata

The Salt Lake City was developed as an alternative to accommodate the growing population in Kolkata after independence. The town was developed in the sixties filling up the wetlands of moribund Bidyadhari river. The government commenced the most significant and enormous land use shift in 1955, when it purchased 173.70 acres of land for reclamation expansion, and large-scale reclamation of this marsh began in the 1960s.

By 1968, 36 km2 of the southern Salt Lakes had been reclaimed to such an extent that it has become a city, whereas earlier it was a swamp. Total 12.5 sq.km. area was selected for setting up the planned township in phases. (Bardhan & Chatterjee, 2016)

The Salt Lake City satellite township is in Bidhannagar, Kolkata, West Bengal. The entire township is divided into 71 blocks. These blocks are triangular or rectangular in shape with one or three open spaces in each. Row housing with patches of green is th main feature of the blocks. The blocks follow grid iron patter design.

The study area is defined as the Block BD and CD. The block CD and BD is shown below in the map of Salt Lake City.

Figure 40: Location of Salt Lake City

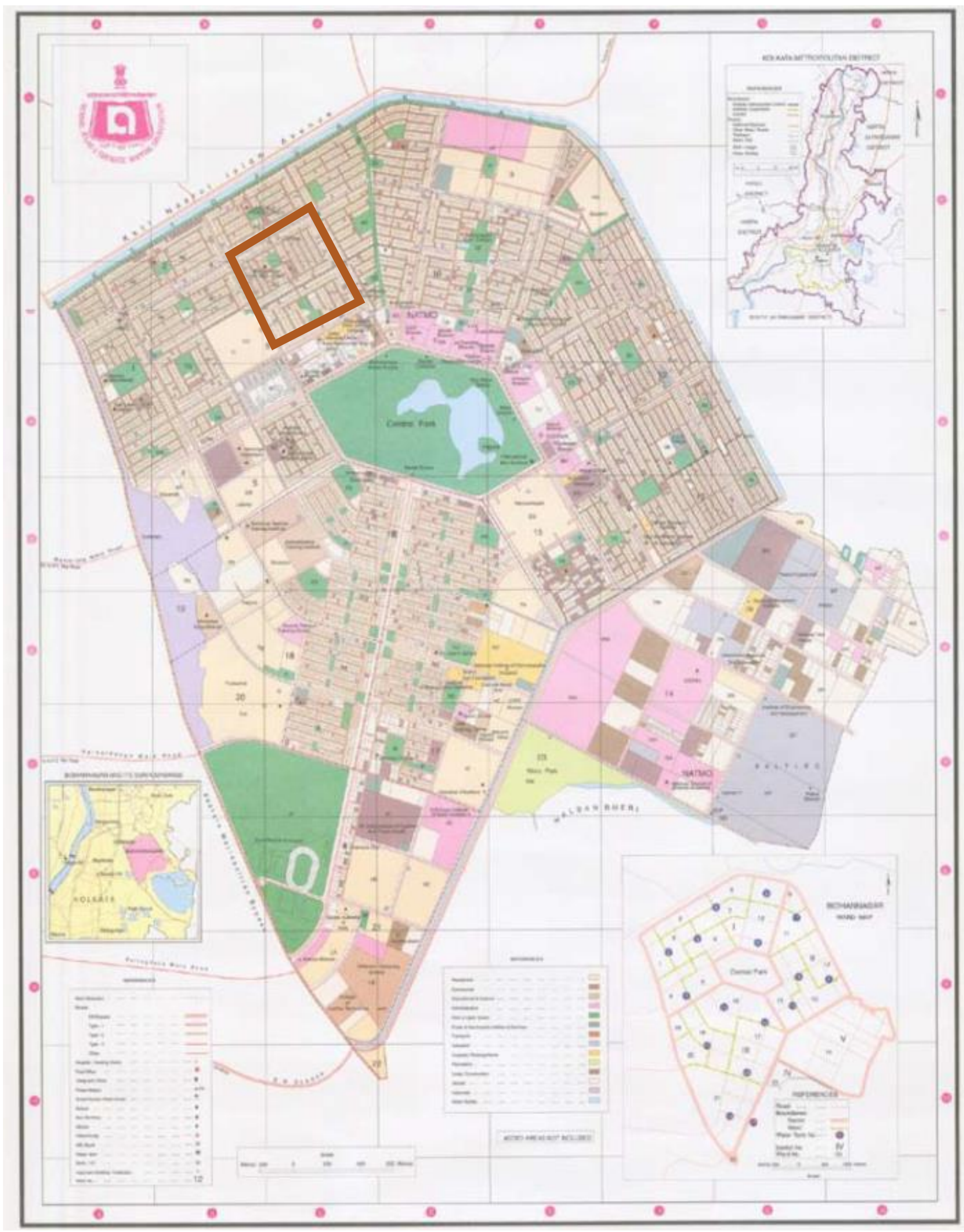


Figure 41: Map of Salt Lake City, Kolkata with study area marked. (Block BD and CD)

3.2 Location of Blocks CD and BD – Study area

The Blocks BD and CD have been taken as the study area. The reason of selection of these two blocks is the shape of the block, which is a perfect rectangle and, the grid iron patter road layout. The blocks BD and CD are located at the Northwestern part of the Salt Lake City.

Figure 42: Sector Map of Salt Lake City showing Block BD and CD.

Figure 43: Satellite Map of Block BD and CD.

3.3 Figure Ground Map

The Figure Ground Map of BD Block and CD Block is represented below. The map shows that the neighborhood has a Fine-Grained structure with row housing and Grid Iron Road pattern. There are three open spaces in the Blocks BD and CD at center. The total number of residential buildings of the neighborhood is 378, and 3 open spaces.

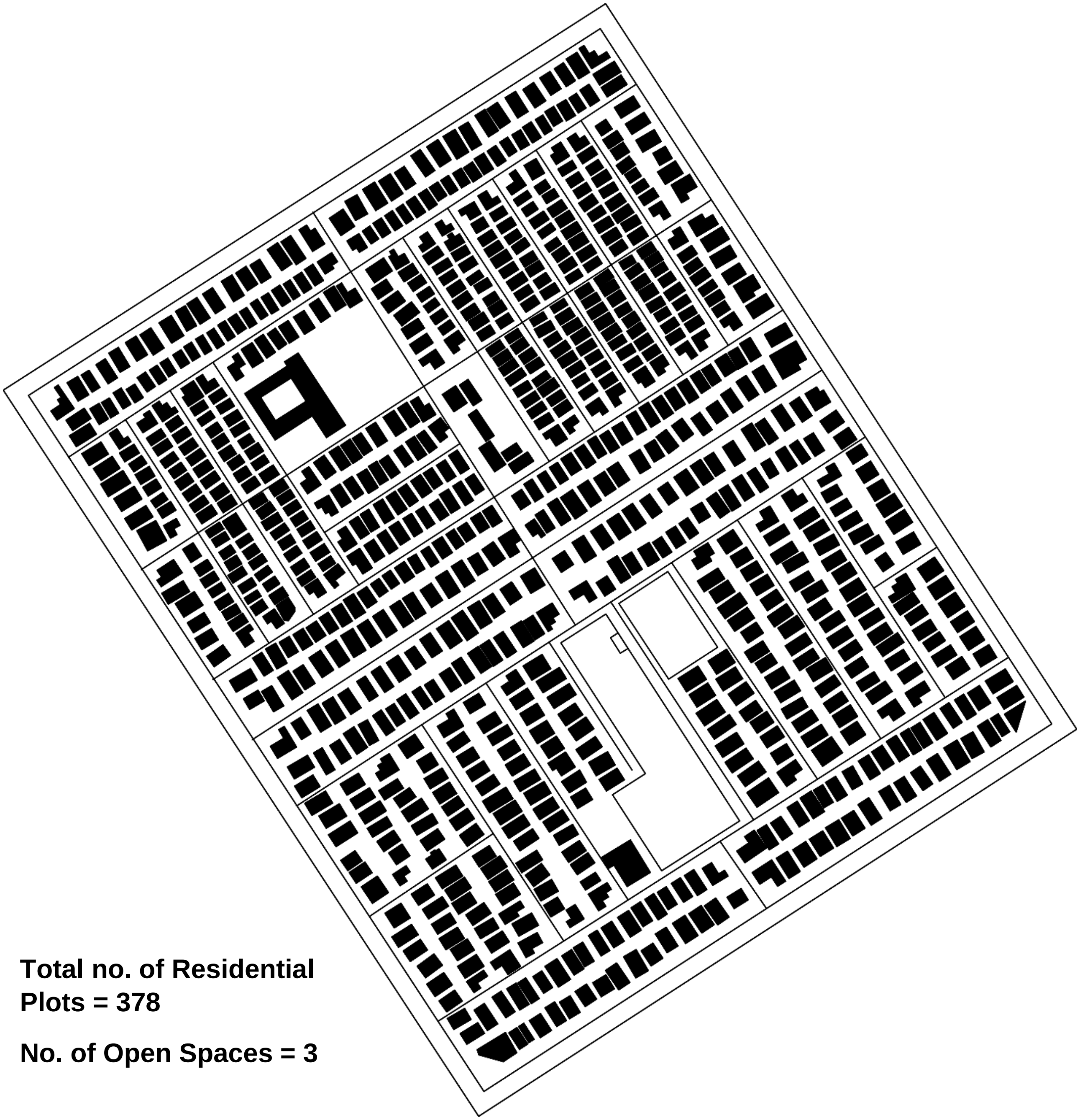


Figure 44: Figure Ground Map

3.4 Land – Use Map

The Neighborhood BD block and CD Block have a mixed-use typology of Land usage. The neighborhood has a combination of commercial and residential spaces with two institutional usages of schools. The total residential land-use is 45%, commercial is 14%, open space is 9% and Roads is 32%.

Figure 45: Land Use Map

3.5 Block Spatial Design Analysis

The BD and CD blocks of the Salt Lake City have a dimension of 646m x 533m.

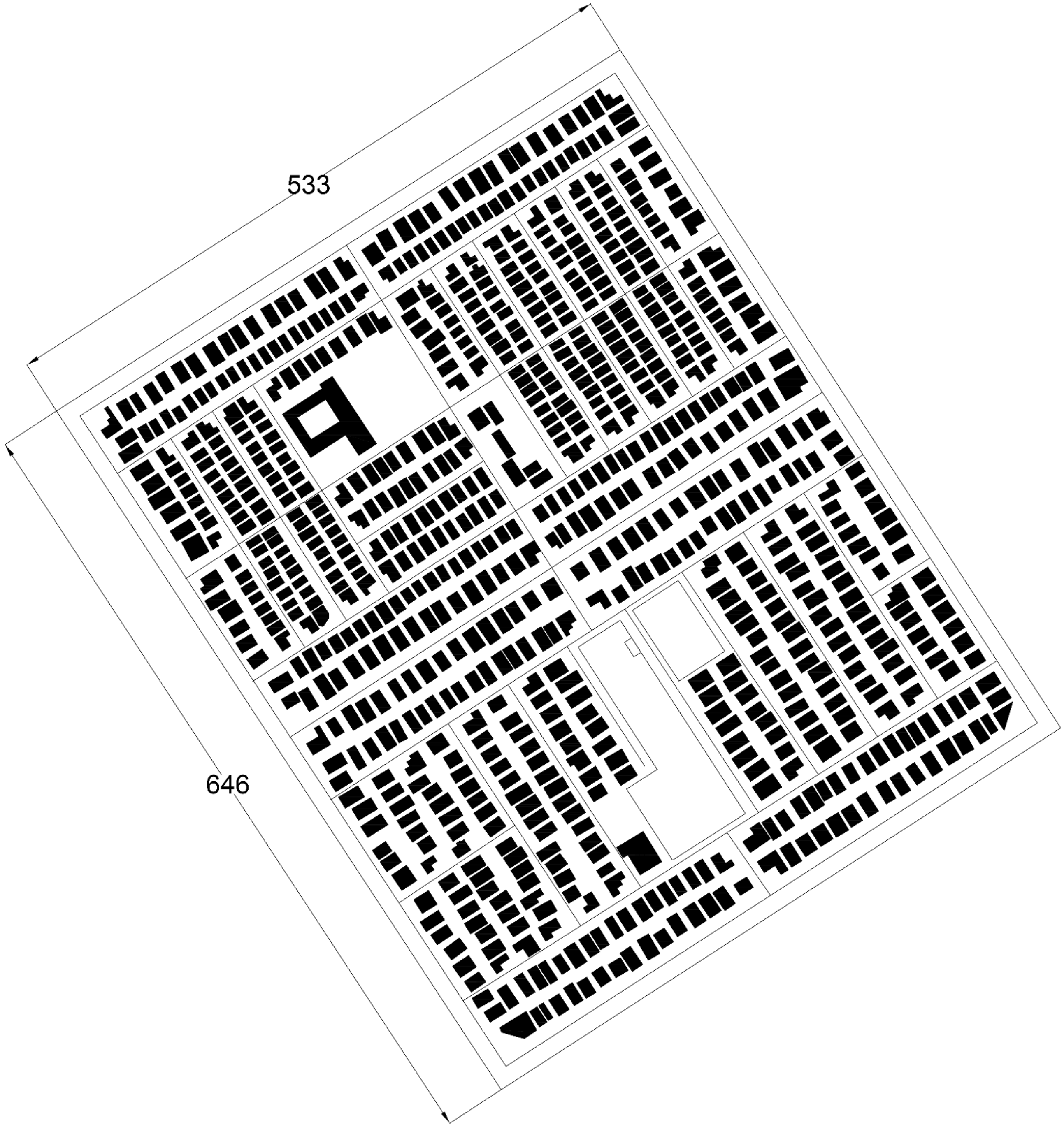


Figure 46: Block dimensions

The Grid Iron Pattern Road network is shown below, There are 7 Major Roads that cut the site from West to east and 1 major road that cuts from North to south. All the other roads are local streets. The width of roads is 9m and road covers 34% of the site area.

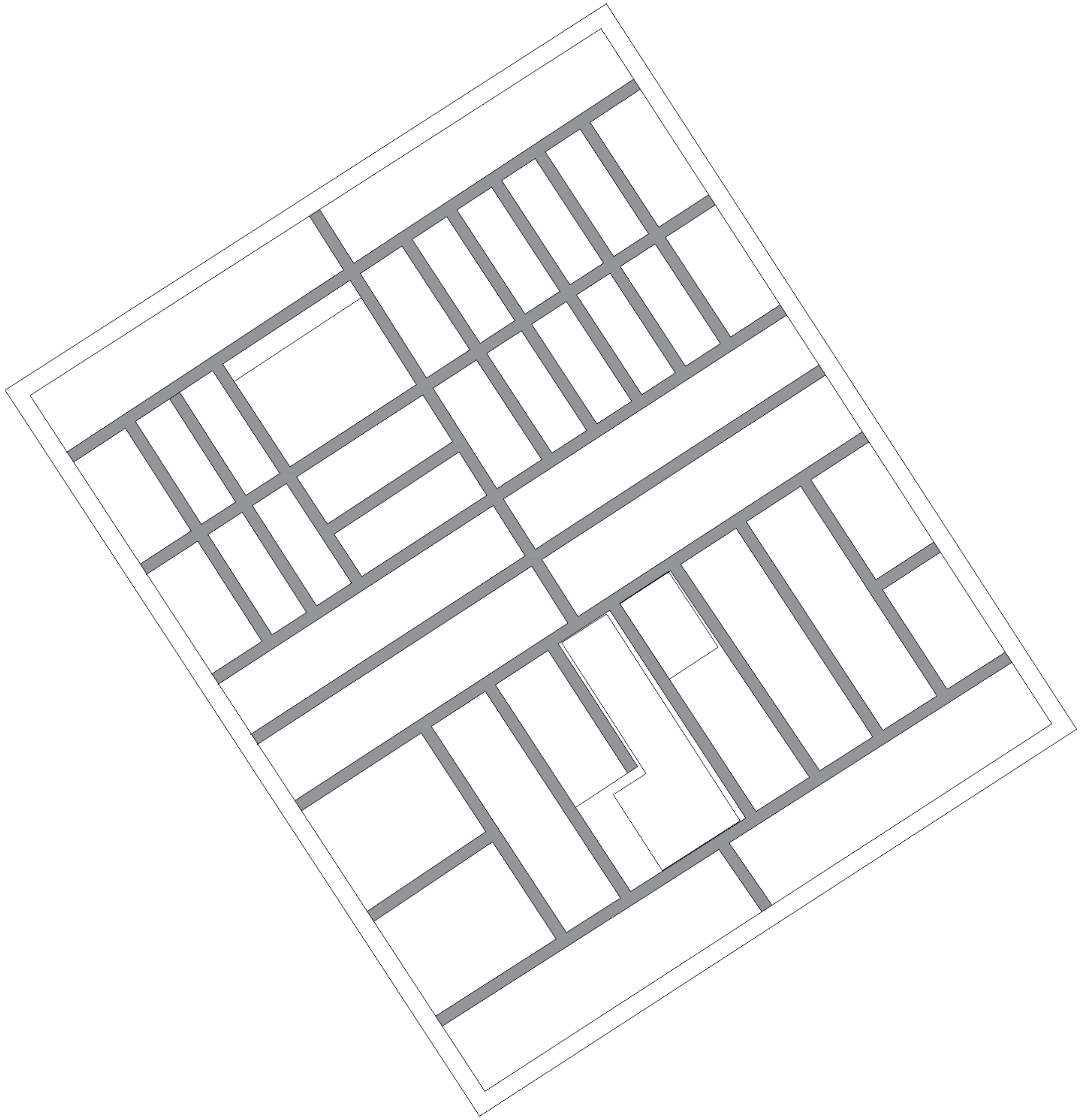


Figure 47: Neighborhood Road network

The Road network divides the site into 38 Secondary Plots which are then further divided to form the individual row housing blocks.

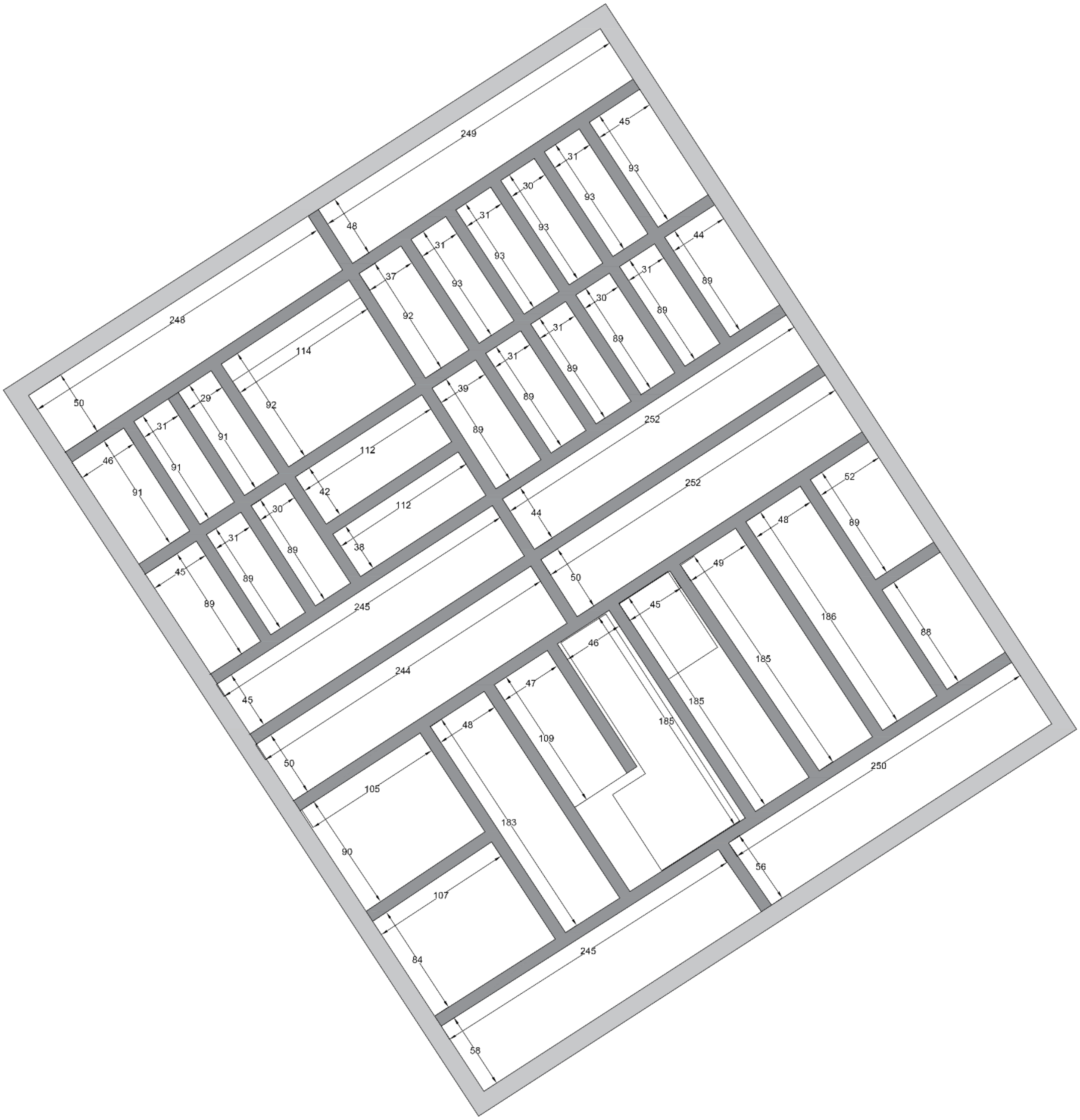


Figure 48: Secondary Block dimensions

3.6 Amenities

The site has three basic amenities – Open spaces, School and Market complex.

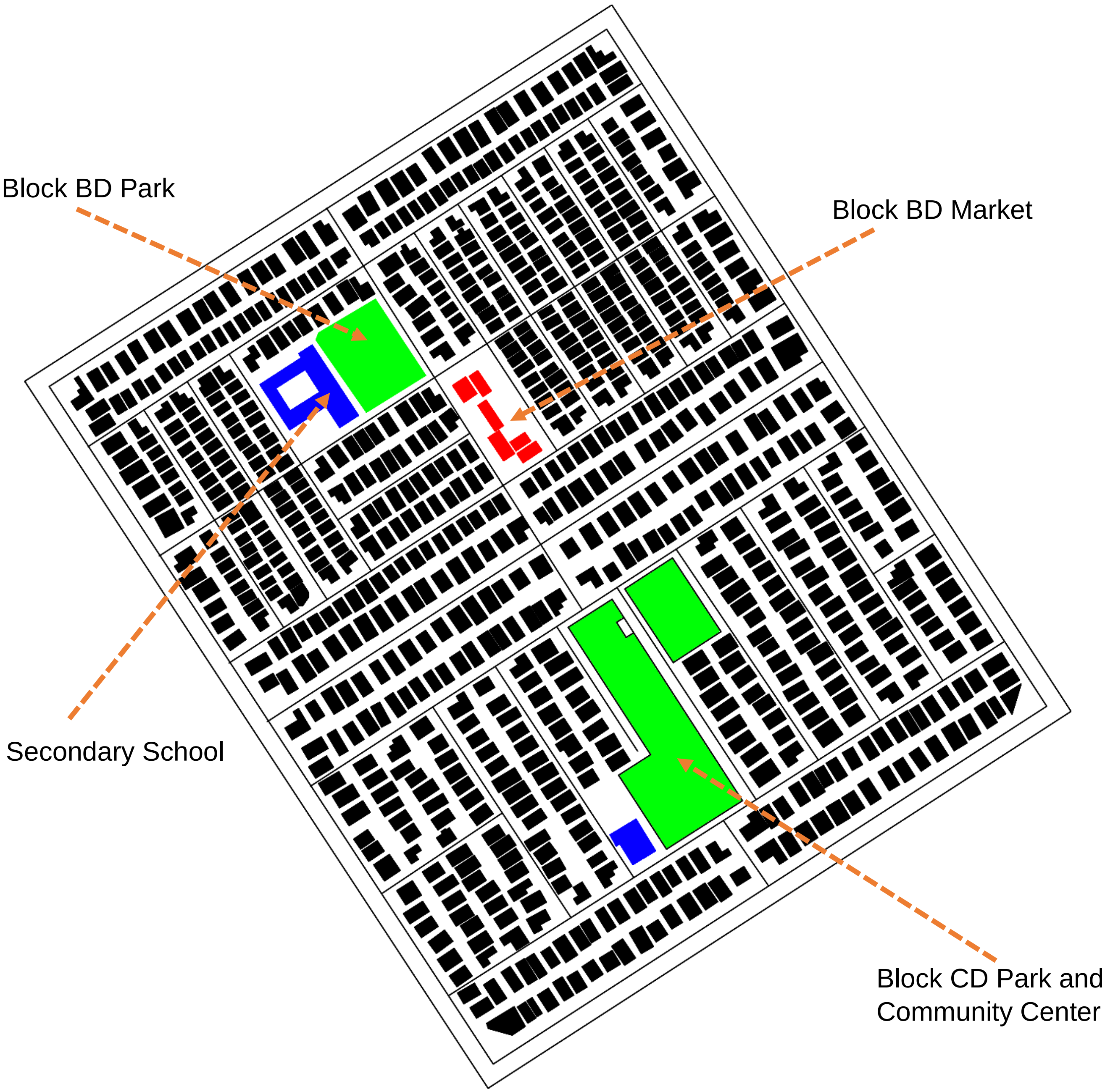


Figure 49: Amenities Location

Maximal Radial Distance to Amenities

Maximal Radial Distance to Park = 283m, School = 592m.

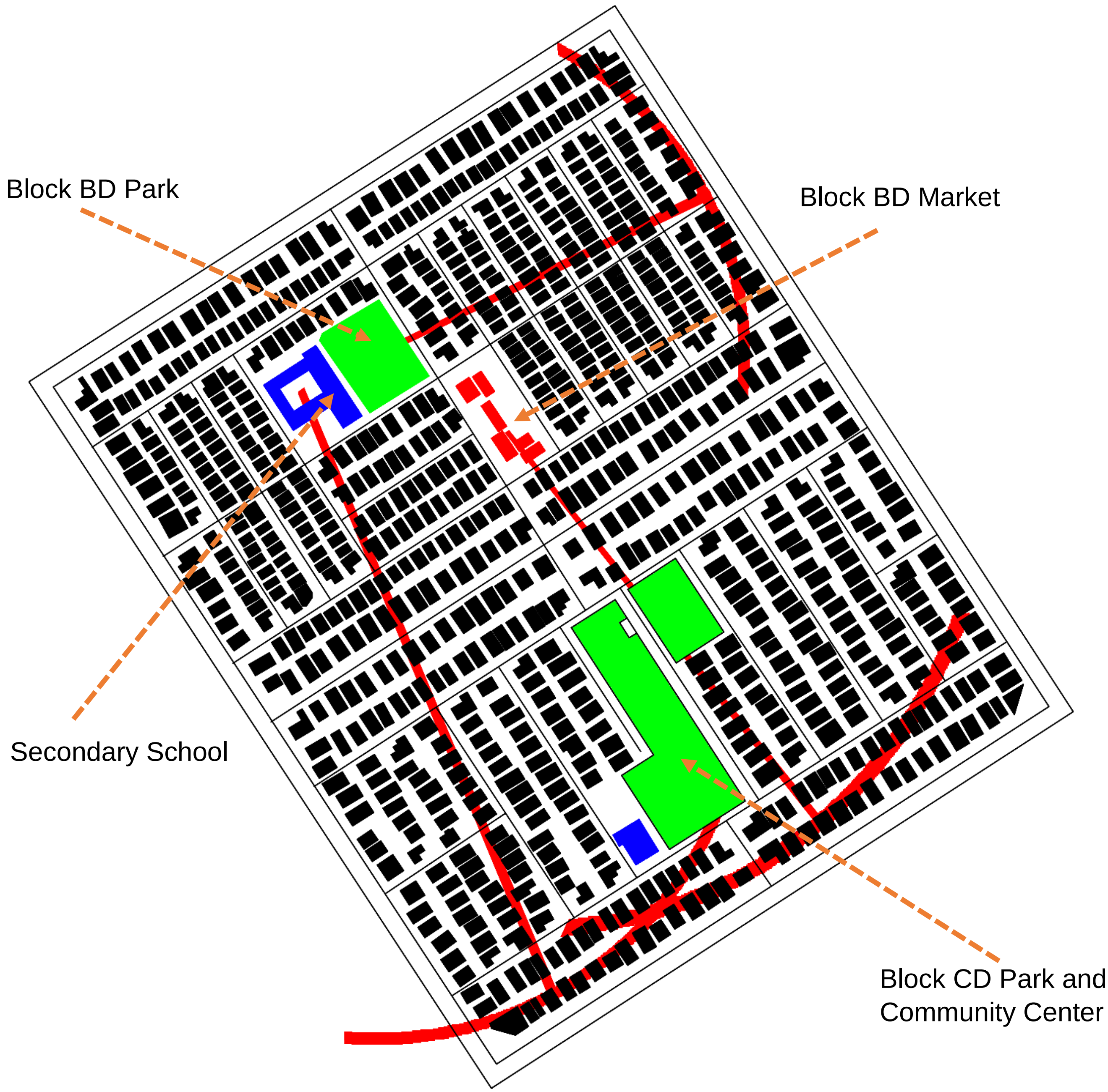


Figure 50: Amenities Location

3.7 Shading Analysis

On a typical summer day, 26th June, 2pm., The 17% of the road surface receives shadow.

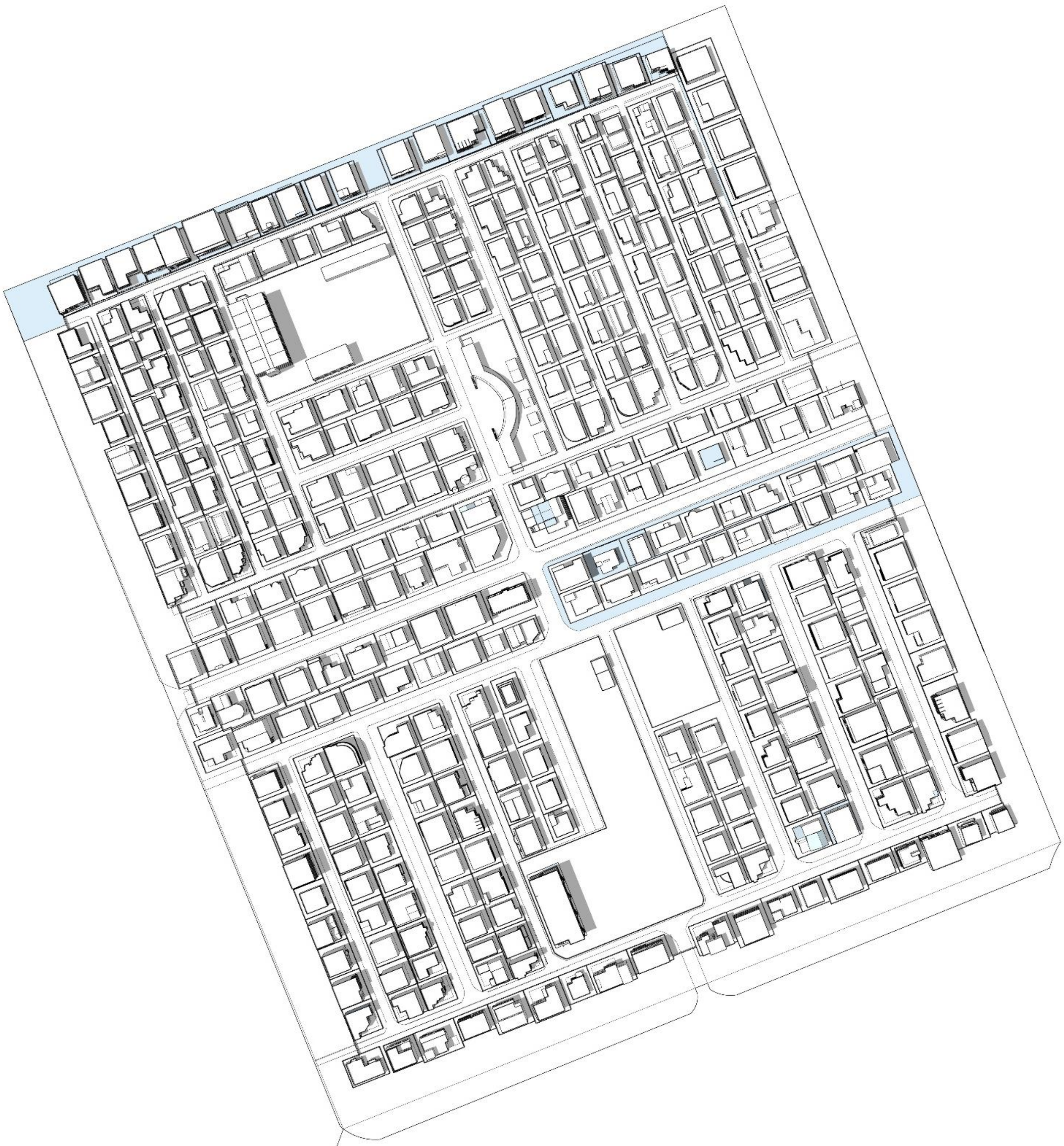


Figure 51: Site Shadow Analysis

3.8 Summary

The site analysis has been done on the parameters that would be optimized using the Generative Design algorithm.

The existing site parameter values include:

a. No. of Plots = 378
b. Site Open Space Area Percentage = 9%
c. Area Covered by Roads = 34%
d. Maximum Radial Distance to school = 592m
e. Maximal radial Distance to Park = 283m
f. Shading Percentage = 17%

This page is intentionally left blank.

4. Project Development

The research intends to identify the different parameters and their set of required goals for urban design and spatial layout of a neighborhood and create an algorithm for automated design layouts. These parameters can be quantified through mathematical expressions and rule based geometrical system. The quantified parameters can be used to create a generative algorithm that outputs different automated design layouts of the neighborhood. These design options can then be assessed for the optimal solutions.

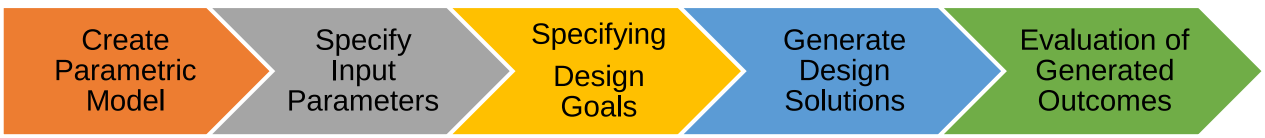


Figure 52: Methodology for Generative design Algorithm

The first step involves the creation of a rule based geometric model of the neighborhood that responds to change in layout based on the parameters the user inputs. This parametric model is developed based on the different Urban design Theories as the logic.

After the parametric model is created, certain Input parameters that change the design of the outcome need to be fixed along with the Project goals that will later be used to analyze the design outcomes.

After the project goals are defined, the parametric model is now ready for generative design study. The input parameters and the type of generative outcome required is specified. The Generative Design tool of Autodesk gives simulates the different scenarios and showcases the results.

The Generated outcomes are then evaluated based on their scoring criteria for different project goals. A detailed approach of the project is explained in this chapter.

4.1 Parametric Model

"Parametric models used in design are composed of a variety of modules that combine computation with geometric operations, none of which are easily differentiable." (Nagy D. , Learning from nature, 2017a)

The parametric model is the backbone of the generative design study. The parameters of the Model represent the design variables that serve as inputs for the Generative Design. The optimization study finds a set of values that produce the best performing design.

The extent of the explorable design space is determined by the parameters and their respective value ranges. They are picked and defined by the human designer, who is now tasked with expressing an abstract multidimensional notion rather than designing a particular item. The parameters should be chosen carefully, as too many inputs may result in a design space that is too large to explore, while too few inputs may rule out a potential optimum solution.

In the case of Generative Design for the Salt Lake City Neighborhood, two different types of inputs are differentiated: parameters with variable and fixed values. The fixed parameters are user specified and can be changed.

The fixed parameters include:

- Site Boundary
- Road Widths (Local Street and Through street width)
- Plot Dimensions (Length and Width) – The individual blocks
- Plot Offset – Offsets of one plot to other
- Amenity Area (School) – The total plot size for the school.
- Building heights – Individual unit heights.

The variable parameters include:

- Road Position on X and Y Axis (Through street)
- Number of Roads (Through streets)

The variable parameters are the input parameters to the Generative Algorithm. Their values are changed by the algorithm to generate different design solutions.

The parametric model is created in Autodesk Dynamo which is a Visual programming tool in Revit. The process of creation of parametric model is sequentially described below.

4.1.1 Site Boundary

The site boundary is first modelled in Revit by using Model lines. The angle of rotation, length and boundary is defined. These lines are now selected in Dynamo and converted to curves. Further, a site polygon is created, which is then converted to a Surface.

The boundary of the site is the major parameter that defines the design space. The site surface mesh is created which contains a set of coordinates and divisions.

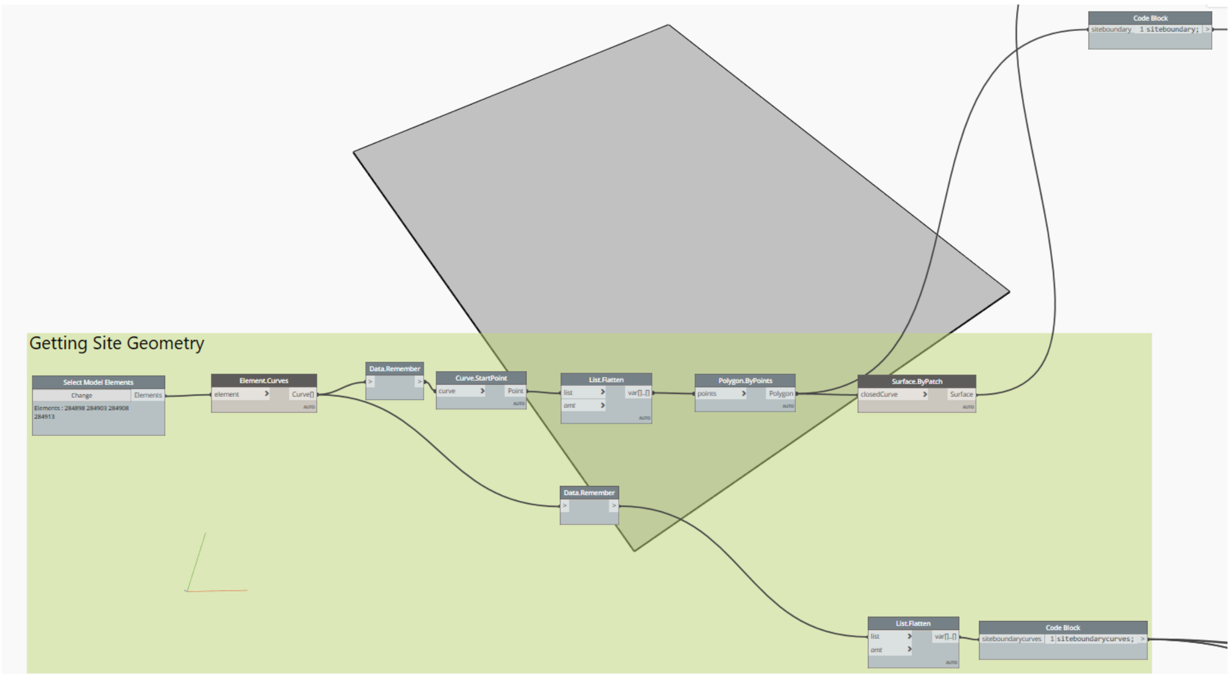


Figure 53: Getting Site Boundary

4.1.2 Primary Roads – Through streets

The site boundary curves, or lines are now divided into the number of parts to create the major roads. A set of points are created on the boundary curves based on parameter. The parameter defines the location of the point. For the start of the line, the parameter value is 0 and for the end point, the value is 1. For when we change the value of parameter between 0 to 1, the point moves from one end to the other of the line.

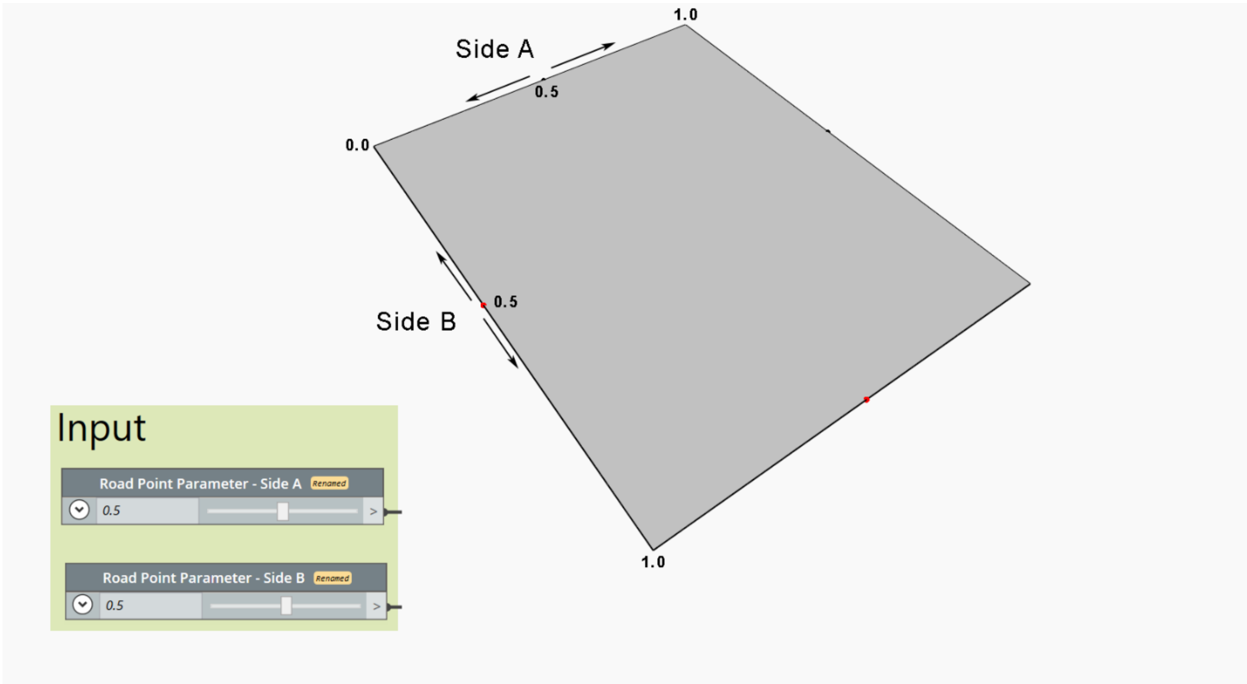


Figure 54: Major Road Location Points

These points are now connected to create the center lines for the Major Roads. The number of roads on the X axis and Y axis is an input parameter defined by the user.

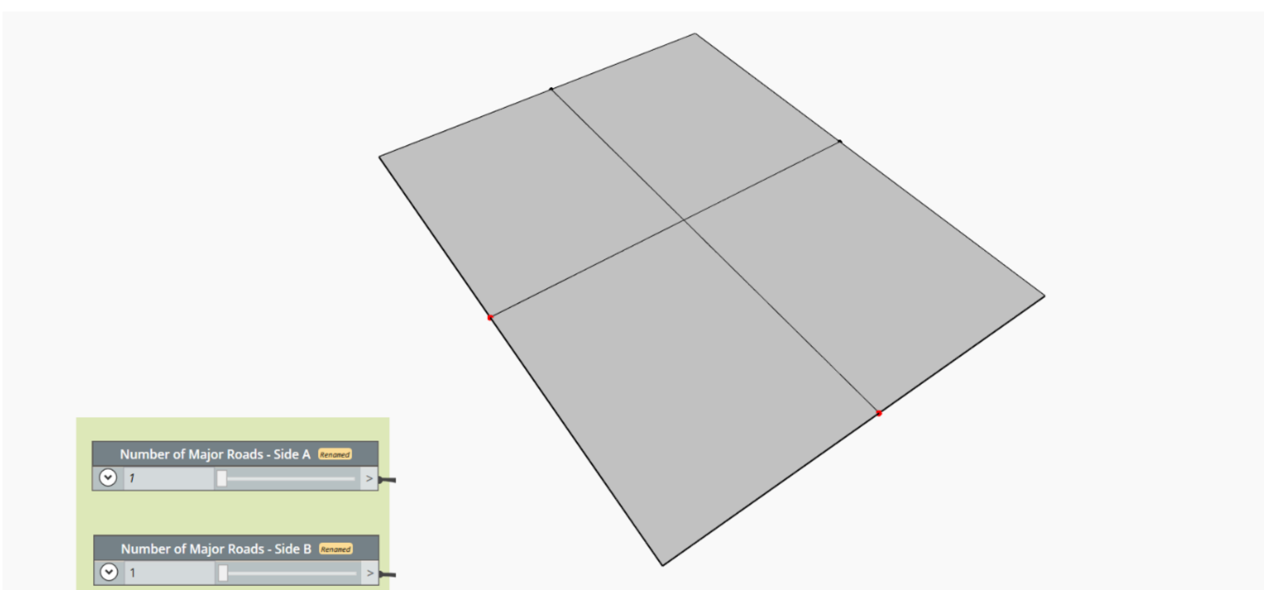


Figure 55: Creating parametric Road center lines

Further, the Road width is added using user input defined parameter.

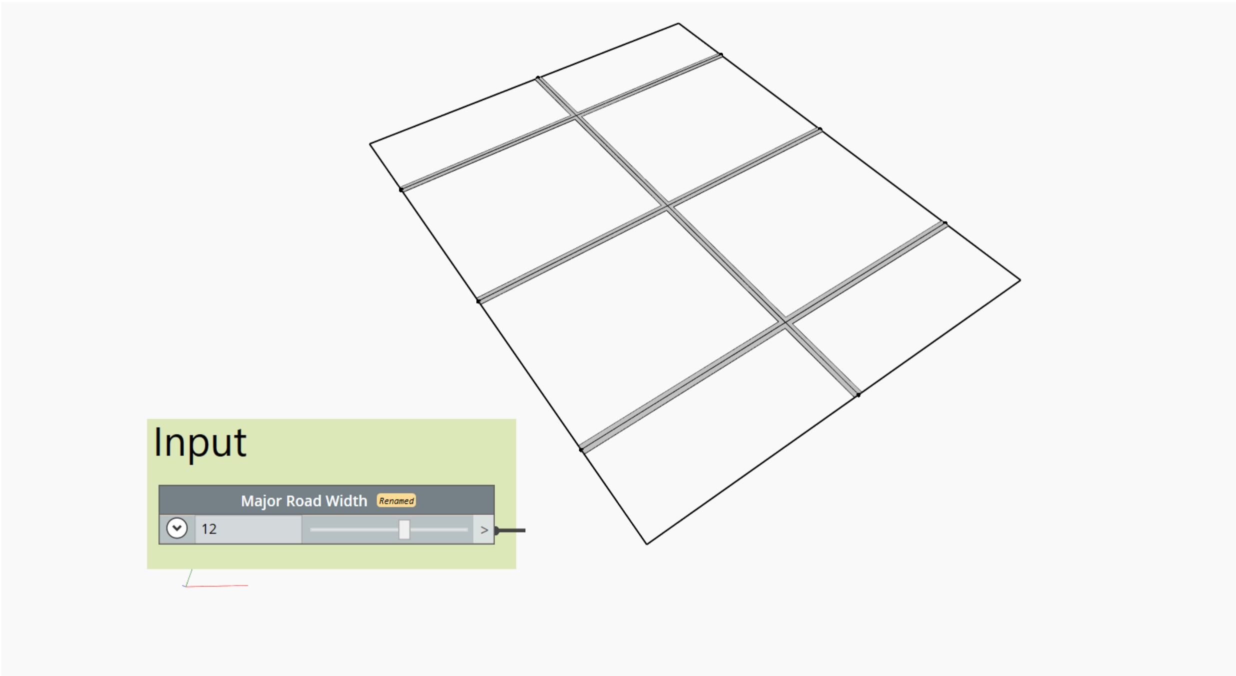


Figure 56: Adding Road Width

4.1.3 Major Plots

The Road surface is split with the Site Surface to create the Major Plots.

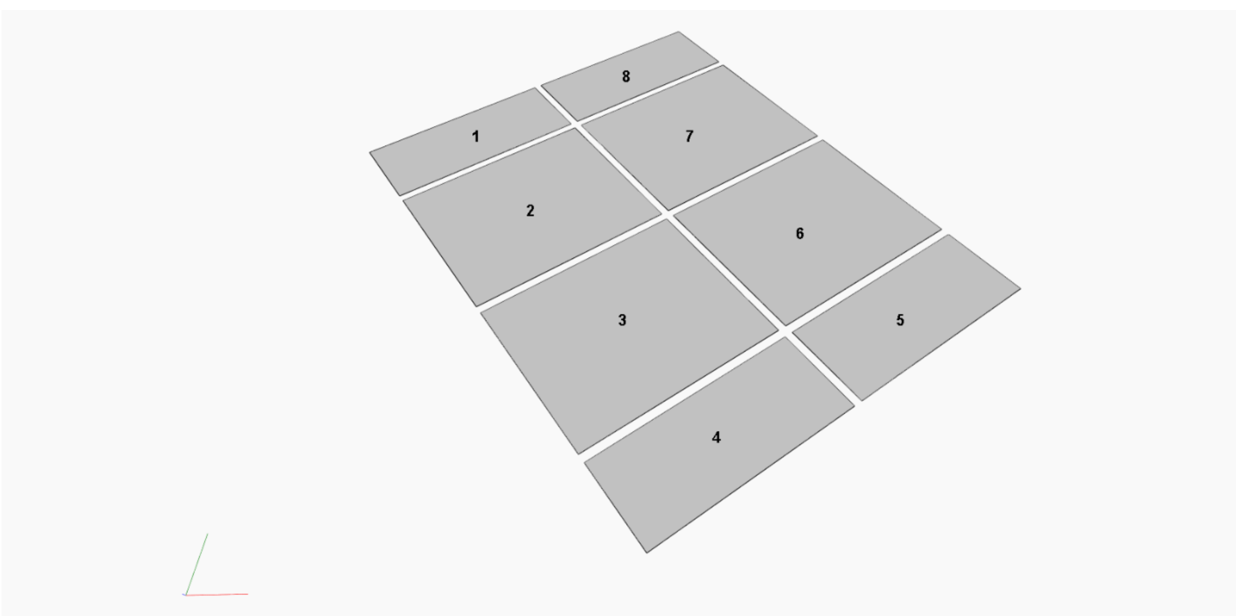


Figure 57: Splitting Site Geometry with Road Surface to create Major Plots.

4.1.4 Secondary Roads – Local streets

After we have divided the Major Plots, we next find the length of the Perimeter lines of each of the Major Plots. The longer side is now divided at fixed length to create shorter plots. This is done based on the Grid Iron Pattern Design theory. Also, if the length of longer side is less that the dividing length, those plots are not cut.

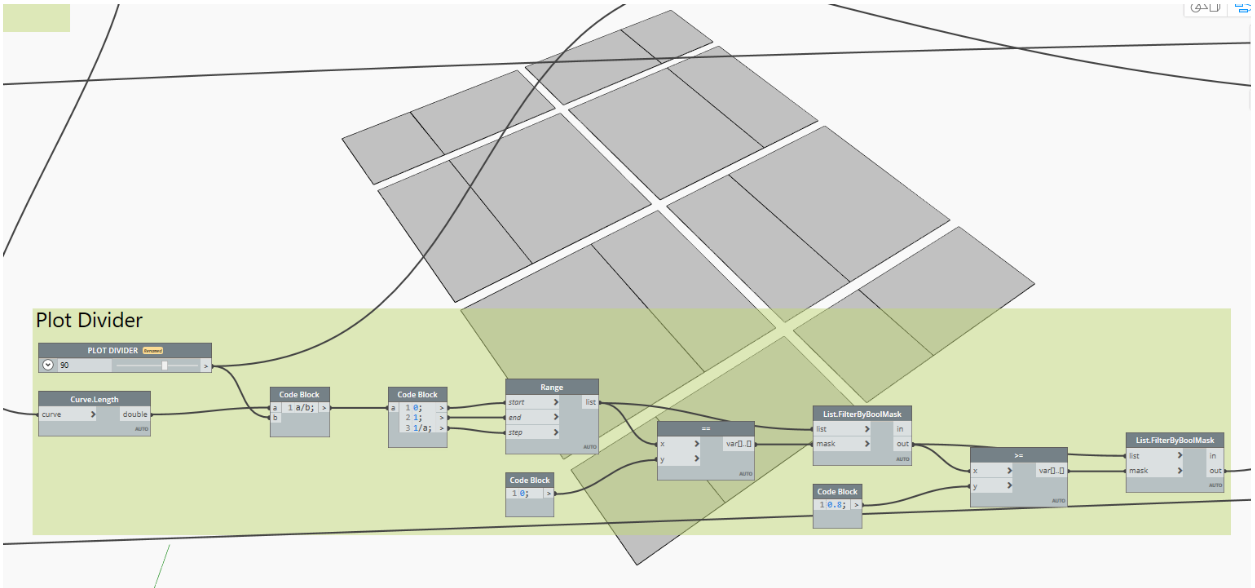


Figure 58: Major Plot division based on User defined input.

Now, again we find the lengths of the divided plots to find the longer sides. Next, we find the width of the Blocks which are user defined, and we multiply the width of the plot with two and add the width of Secondary roads to further divide the plots into Local Streets. The Individual Plot Length and Local Street Width are user defined parameters.

Offset of Local streets = (2 * Width of Blocks) + Width of Local Street

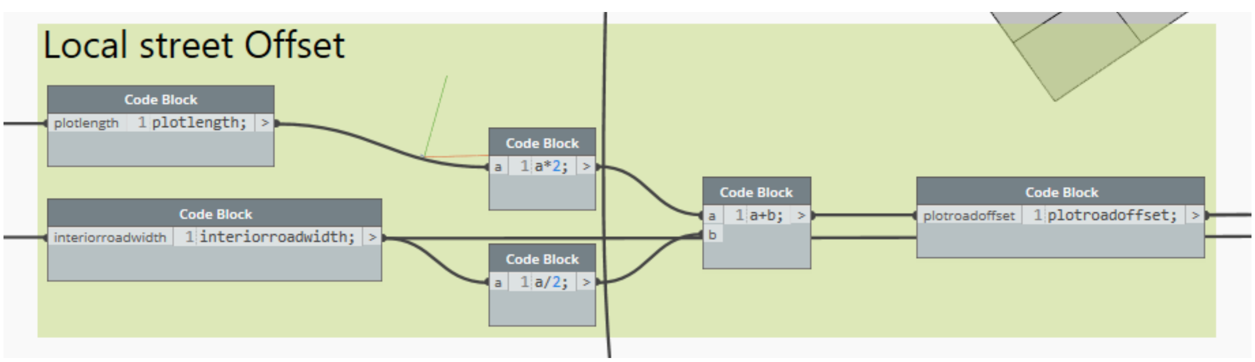


Figure 59: Local Street Offset Calculation

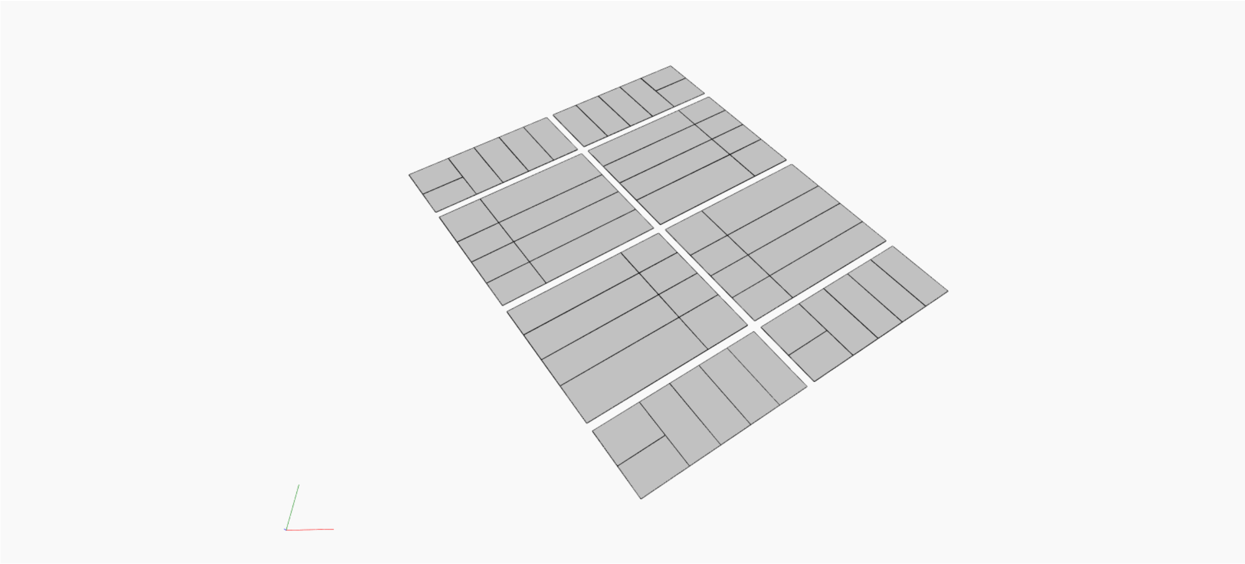


Figure 60: Local Street Center lines.

Next, we add the width of Local Streets to create our Road geometries.

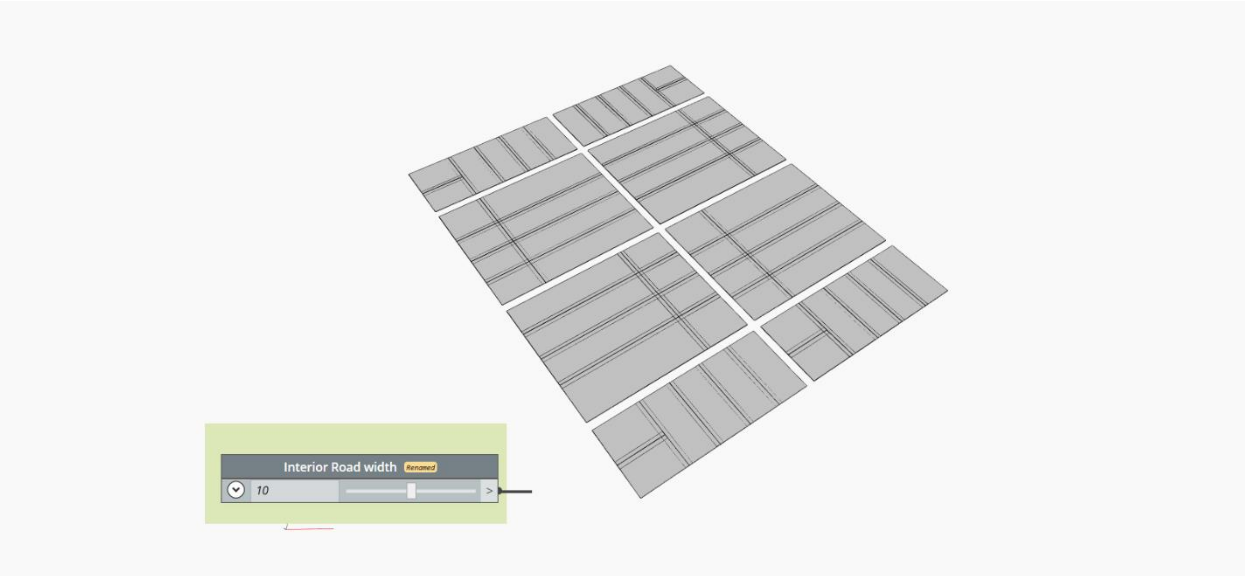


Figure 61: Adding Local Street Width

Finally, we have our road geometries which change with the input parameters of Number of Roads and location of Major Roads.

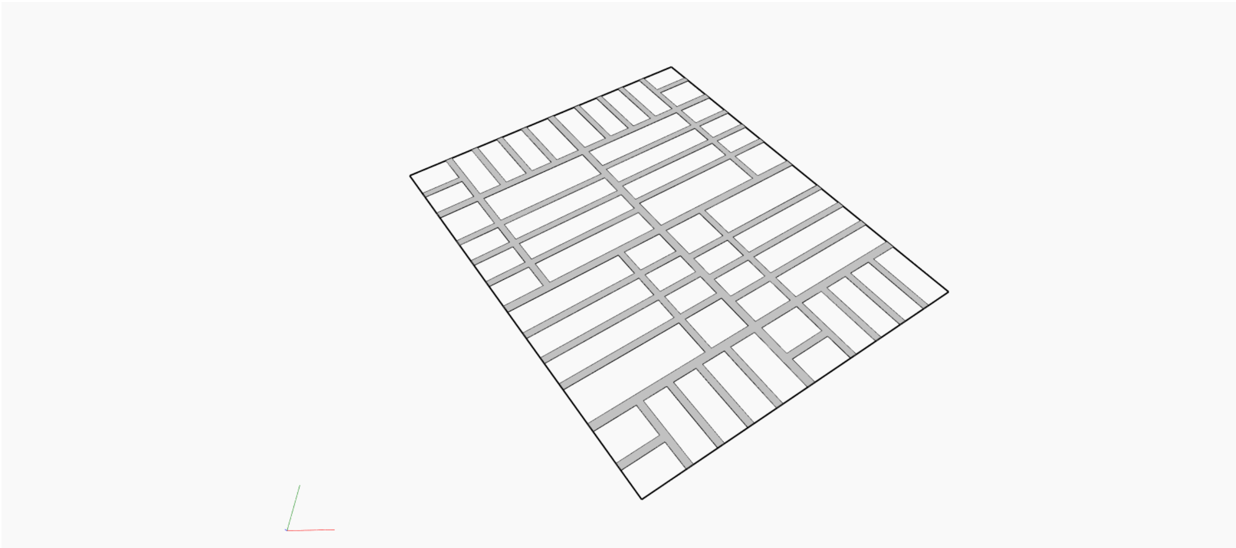


Figure 62: All Roads defined.

Based on the Input Parameter of Location of Major Roads, the local streets change, creating different variations of design.

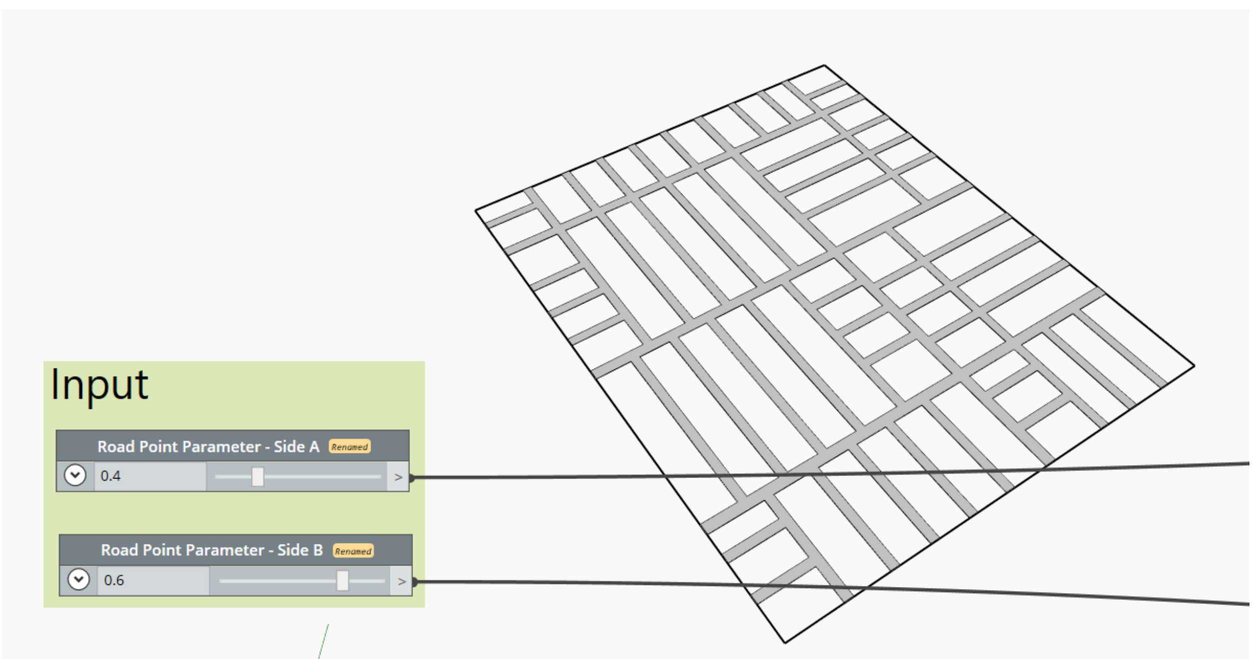


Figure 63:Road Variation 1

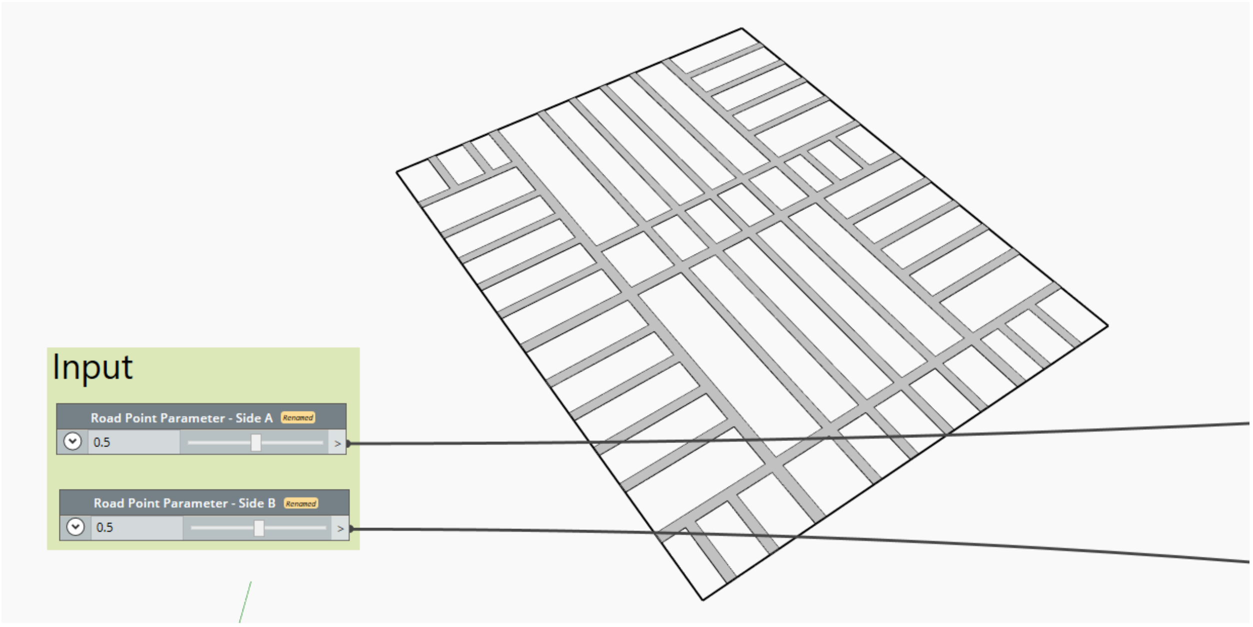


Figure 64: Road Variation 2

4.1.5 Secondary Plots

The road network is now split with the Site Surface to create the Secondary plots with the thickness of two times the length of individual smallest blocks.

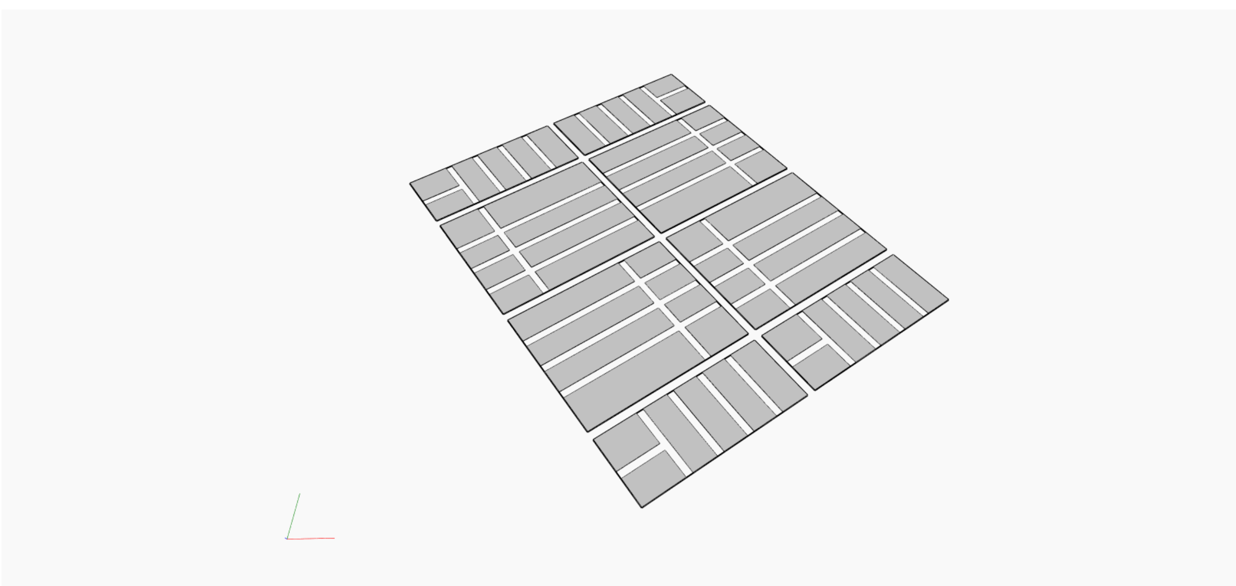


Figure 65: Creating Secondary Plots

4.1.6 Amenities

Two amenities have been included as the scope of the project.

- Secondary School
- Gardens/ Parks

The Gardens/ Parks are located at the center of each Major Plot and the School is located at the center of the site.

4.1.6.1 Secondary School

The Secondary school is located at the center of the neighborhood. First, we find the center of the site and then we find the closest secondary plot to the center of the site.

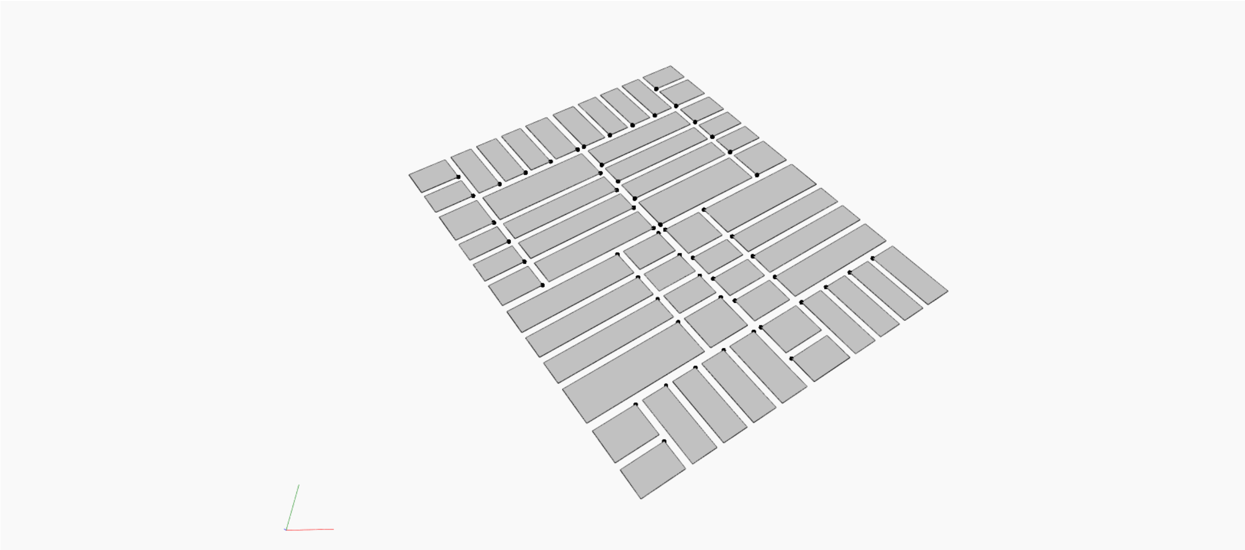


Figure 66: Finding closes secondary plot to the center of the site to locate the school.

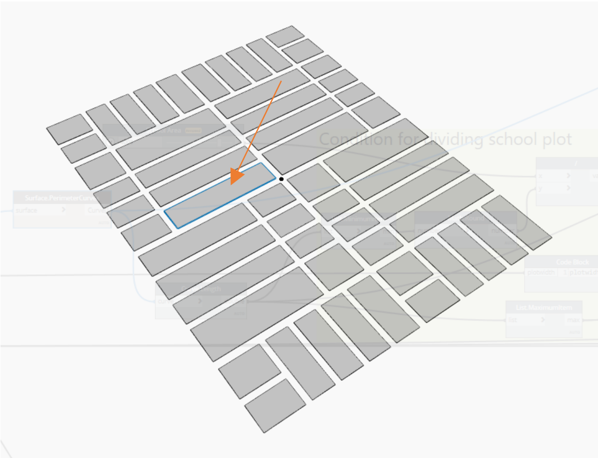


Figure 67: Closest Plot to center of Site

Next, we define the Input Parameter of School Plot area. In the project, the school plot area has been taken as per the existing site conditions. (4000 sqm). Then, we divide the plot and find the school plot boundary and create the school building.

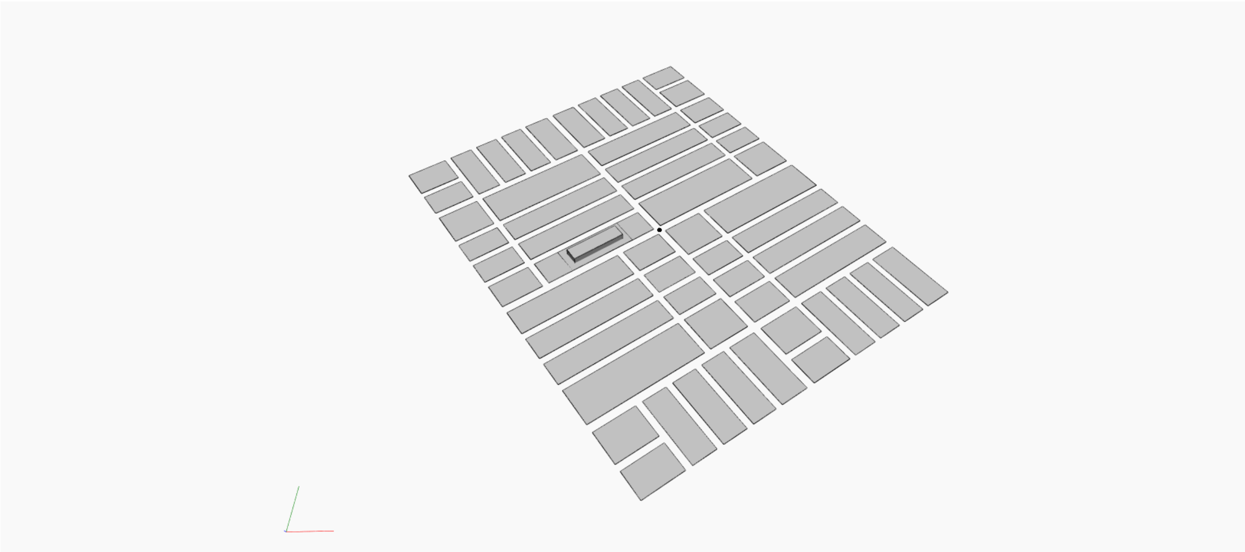


Figure 68: Defining School area and creating school building.

4.1.6.2 Garden/ Park Spaces

The Gardens/Parks are located at the center of the major plots. We find the center of each major plot and find the nearest secondary plot to locate the garden spaces.

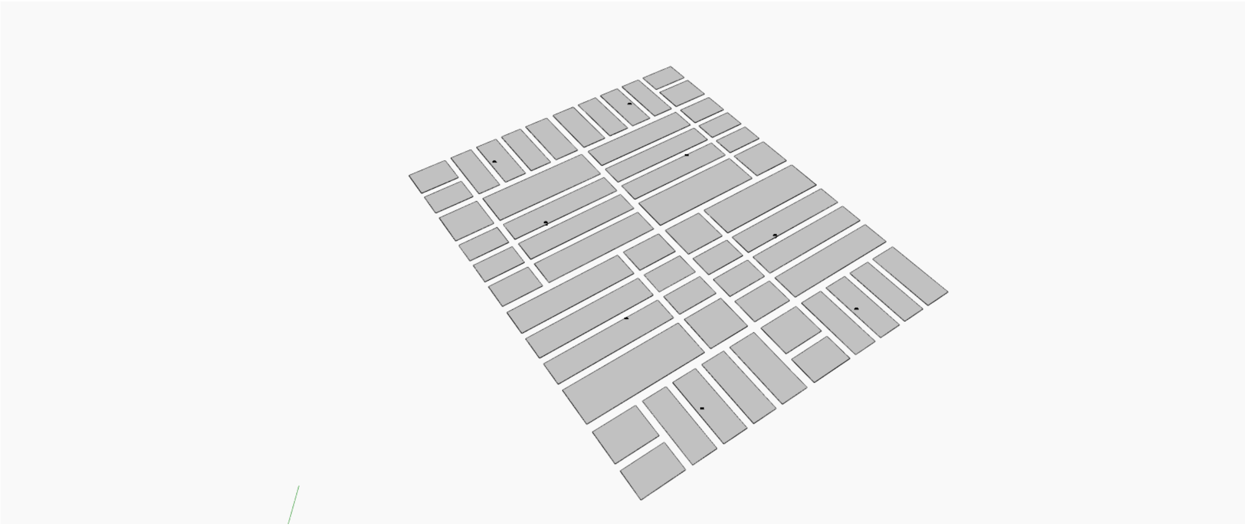


Figure 69: Finding center points of major plots

Next, we find the shortest distance to the Major Plot center point to the secondary plot.

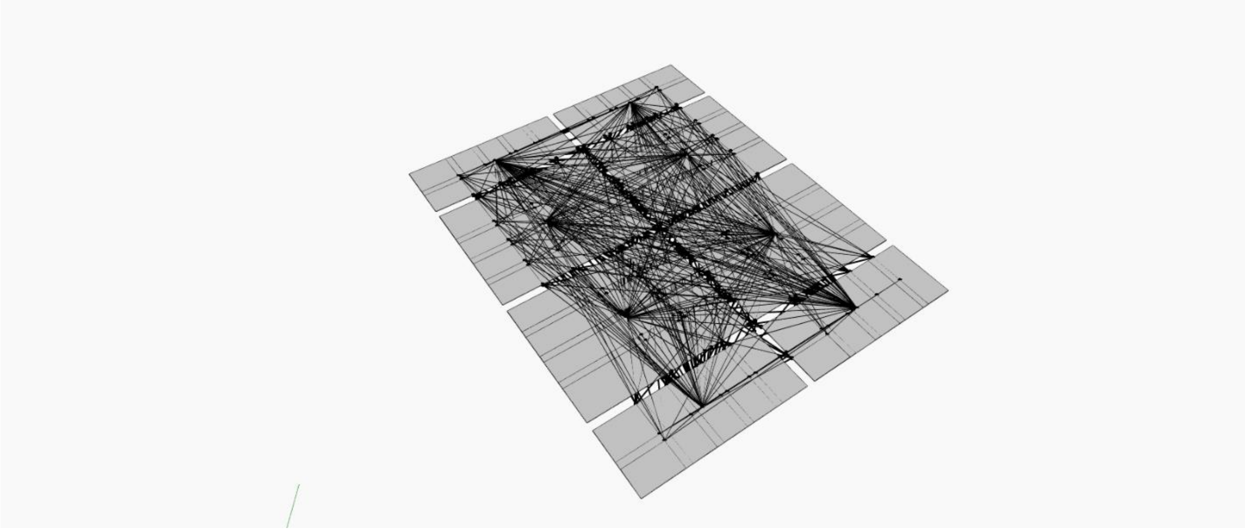


Figure 70: Finding Closest Secondary plot to center of Major Plots

This provides us with the Garden Plots located at the center of the Major plots.

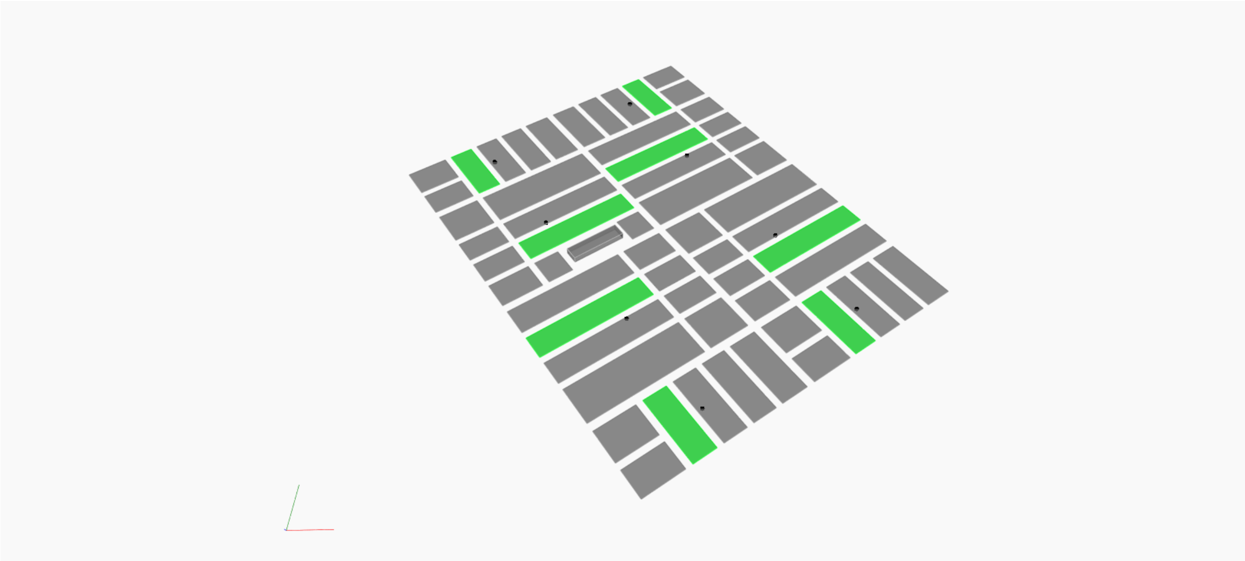


Figure 71: Garden Plots

4.1.7 Building Blocks

Finally, we divide the secondary plot from center to create the individual plots. And then, we add height to the buildings to create our parametric neighborhood model.

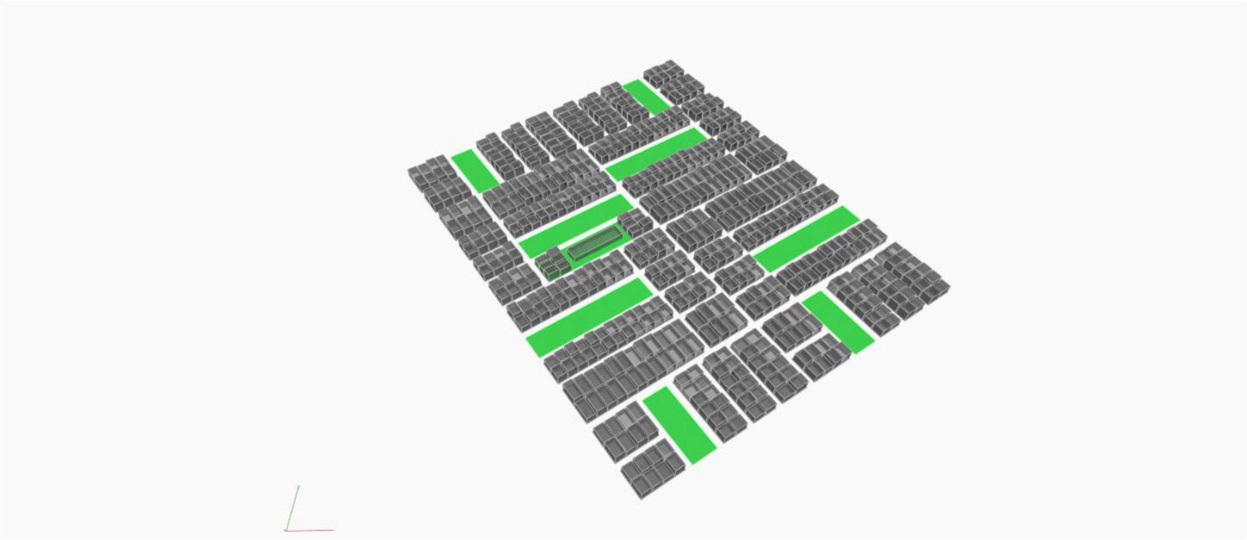


Figure 72: Final Neighborhood Parametric Model with building blocks.

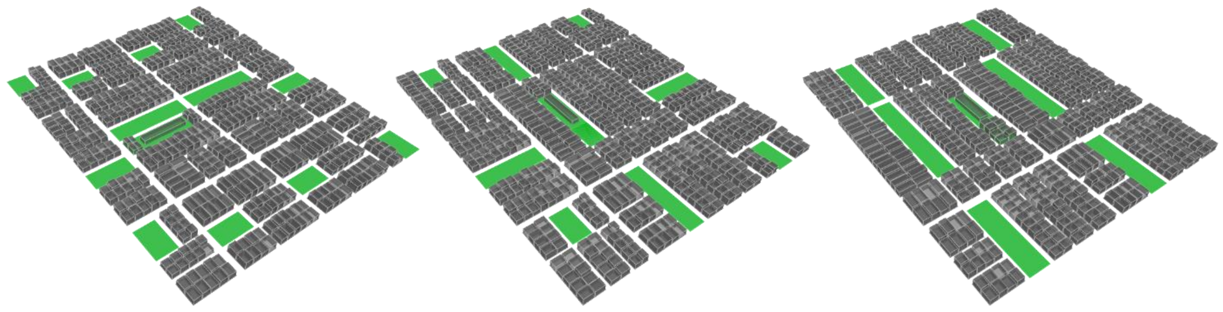


Figure 73: Three versions of the Output with different parameter values of Number of Roads and Location of Roads.

Hence, we are able to create a Parametric Model of the Study Area that corresponds to changing input variables and creates different design options.

4.2 Input Parameters

Two types of Input Parameters have been defined in the Generative Algorithm.

a. Fixed Input Parameters

The fixed input parameters are used as the constraints or the fixed values. These are user specific parameters. The fixed input parameters of this study include:

- Site Boundary
- Road Widths (Local Street and Through street width)
- Plot Dimensions (Length and Width) – The individual blocks
- Plot Offset – Offsets of one plot to other
- Amenity Area (School) – The total plot size for the school.
- Building heights – Individual unit heights.

b. Variable Input Parameters

The variable input parameters are used by the Generative algorithm to generate different design solutions. The generative algorithm changes the values and finds different combinations of the variable parameters to generate different design options.

The variable parameters include:

- Road Position on X and Y Axis (Through street)
- Number of Roads (Through streets)

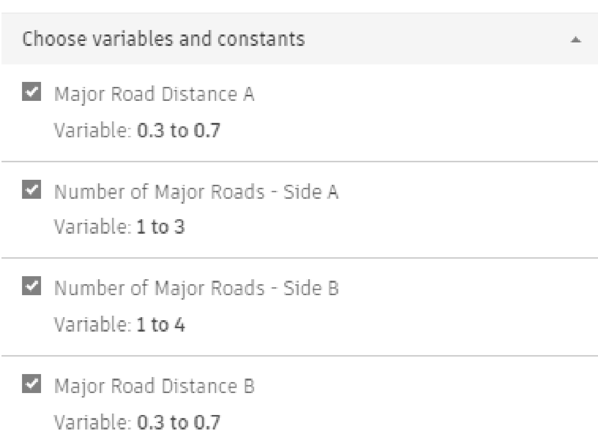


Figure 74: Variable Parameters for the Generative Algorithm

4.3 Design goals

What constitutes a superior design is defined by the design goals. These factors might be subjective to the human decision-maker, such as proximity or comfort, or objective, such as percentage open space or number of plots. Once the design goals have been determined, they must be linked quantitatively with quantifiable indicators. This makes it possible to compare the designs on a level that is objective.

For the Salt Lake City Neighborhood Design, 6 Design goals were identified.

4.3.1 Profit: Maximizing Number of Plots

The first goal of the Design is to create maximum number of Plots. The Generative Algorithm counts the number of Buildings in the Design Option and searches the optimum solutions with maximum number of Plots.

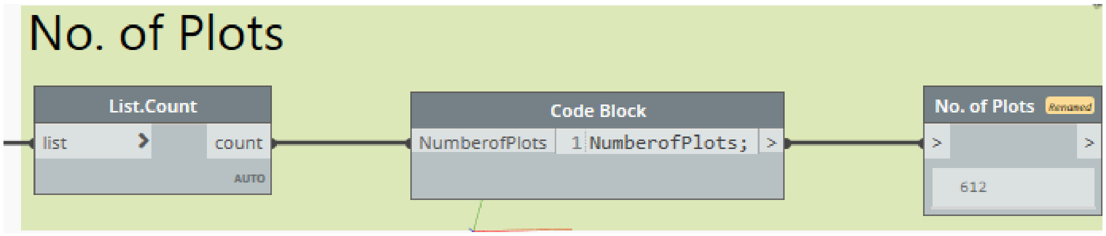


Figure 75: Design goal 1 - Maximizing Number of Plots

4.3.2 Maximizing Percentage Open Space Area

The second design goal is Maximizing the Open space Area. The Generative Algorithm calculates the Total Area of Open Space and finds the percentage of Open Space Area by dividing it by the Total Site Area.

Open Area Percentage = (Total Open space Area/ Total site area) * 100

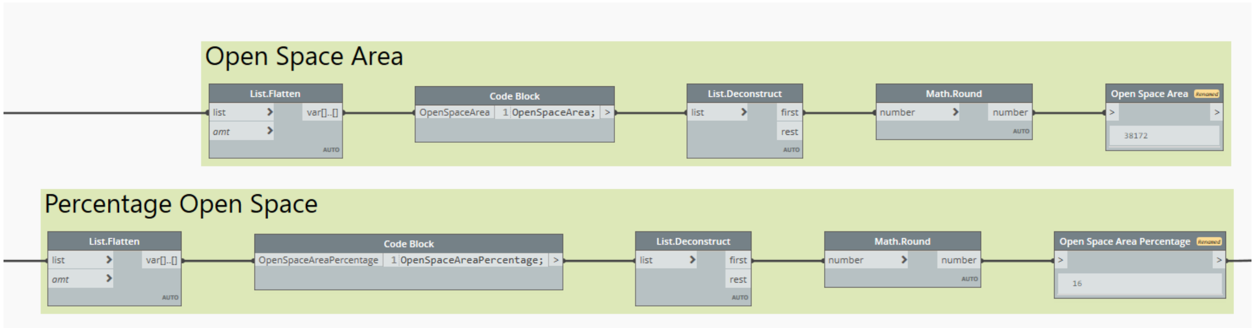


Figure 76: Design Goal 2 - Maximizing Open Space Area

4.3.3 Minimizing Road Area Percentage

The third design goal is to minimize the road Area percentage. The Generative Algorithm calculates the Total Road Area and divides it by the site area to find the percentage of area covered by roads.

Road area percentage = (Total Road area/ Site Area) * 100

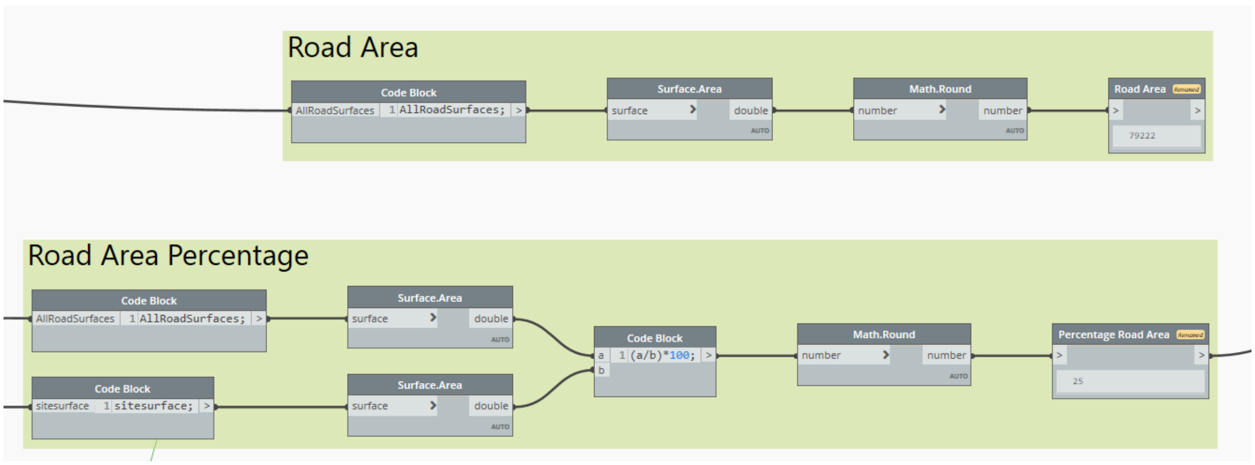


Figure 77: Design Goal 3 - Minimizing Road Area Percentage

4.3.4 Maximizing Proximity to School

The fourth design goal is to Maximize the proximity to school. In order to find the proximity of school to each plot, we create the center points of each plot and the center point of school.

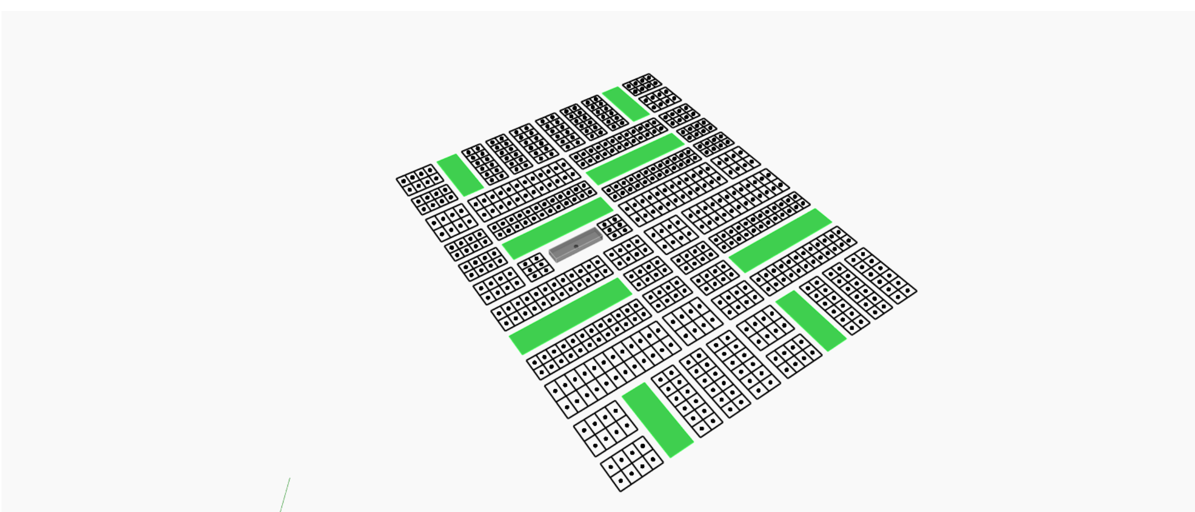


Figure 78: Finding center point of each individual plots and school.

The we create line connecting each plot to the school center. This gives us the radial distance of each plot to the school. Next, we identify the largest distance to school. This largest distance is taken as the proximity parameter, and all the different design solutions are compared based on this distance. The design goal is to minimize this distance.

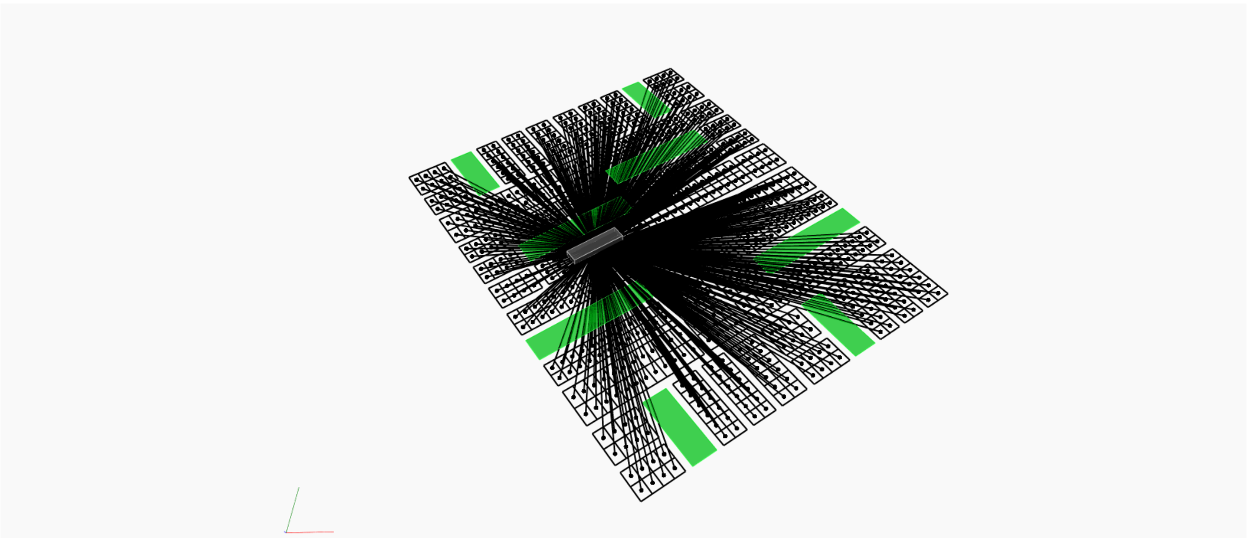


Figure 79: Finding radial distance of each plot to the school.

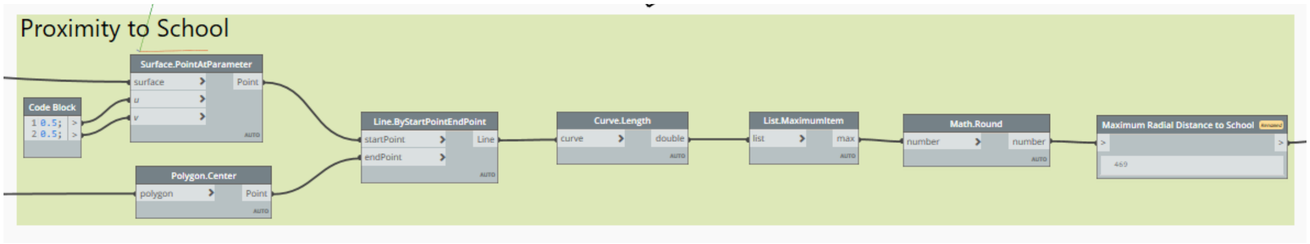


Figure 80: Finding the maximum radial plot distance to school.

4.3.5 Maximizing Proximity to Open Spaces

The fifth design goal is to maximize the proximity to open spaces. The algorithm finds the center point of each plot and the center of each open space. The plots are then grouped together with their appearance in the major plots. The open spaces are located, and the distance of the nearest open space is computed by the algorithm.

The algorithm finds the longest radial distance of a plot to the open space. The goal of the design is to find the minimum radial distance to the open space and hence all the solutions are assessed based on their distance to open space.

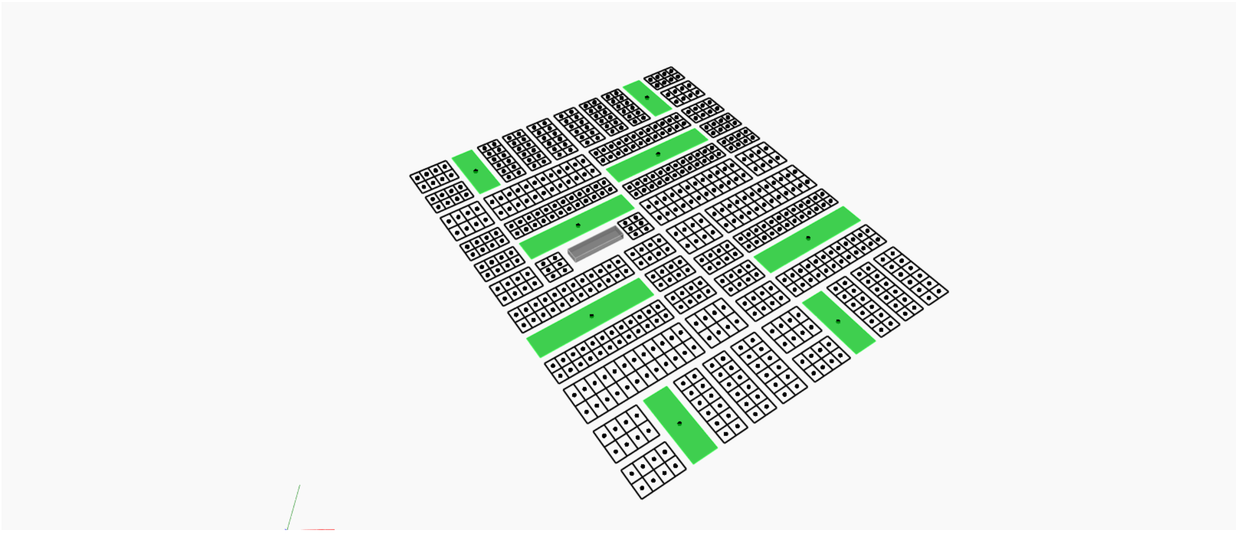


Figure 81: Finding center points of open spaces and all individual plots

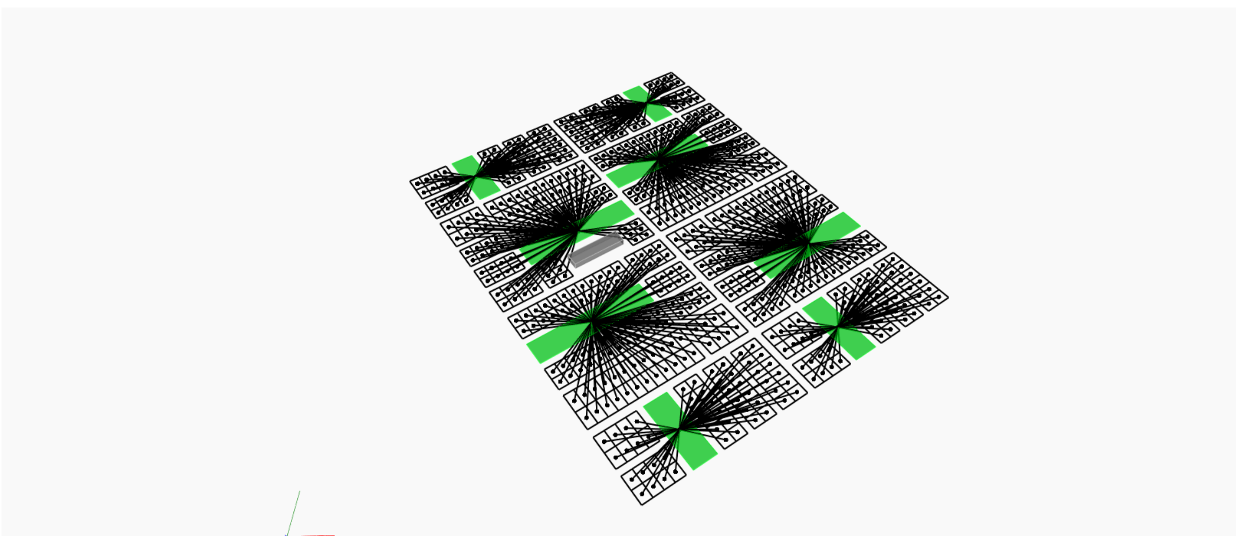


Figure 82: Finding radial distance of plots to open spaces

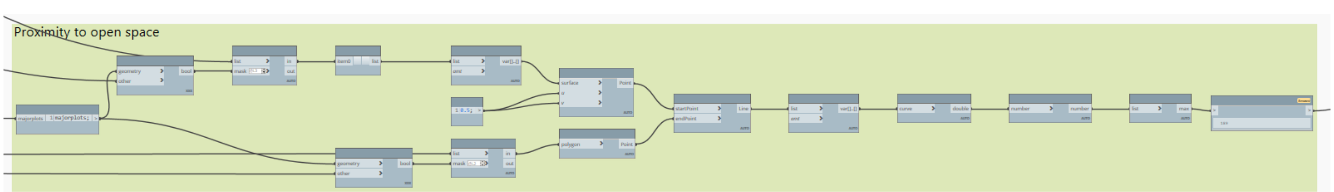


Figure 83: Finding longest distance to an open space.

4.3.6 Maximizing Percentage of Road Shaded

The sixth design goal is to find the solution that has maximum road shading area. The road shading is calculated on a single day and time.

The date and time selected for the shading calculation is: 26th of June, 2:00 PM.The solar angle on this date at the site is calculated and the shadow is cast from each building to the road surfaces. The road surface is now split with the shading surface and the cumulative shading area is calculated.

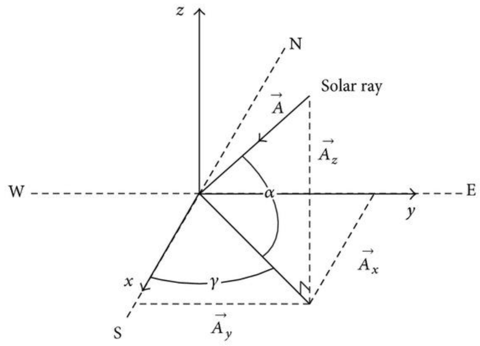


Figure 84: Finding Solar Angle

The shading area is divided by the road area to find the percentage of road shaded. All the design solutions can now be compared for the shading on the same day.

Shading Area Percentage = (Total Road shading area/ total road area) * 100

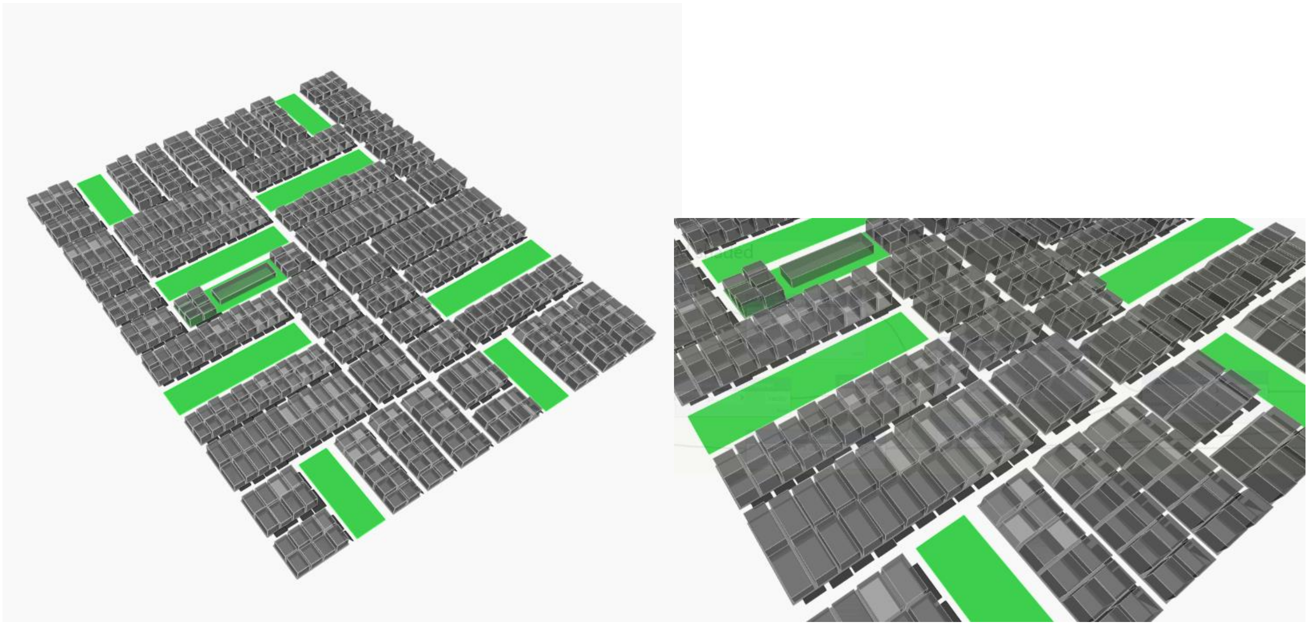


Figure 85: Shadow on road surface

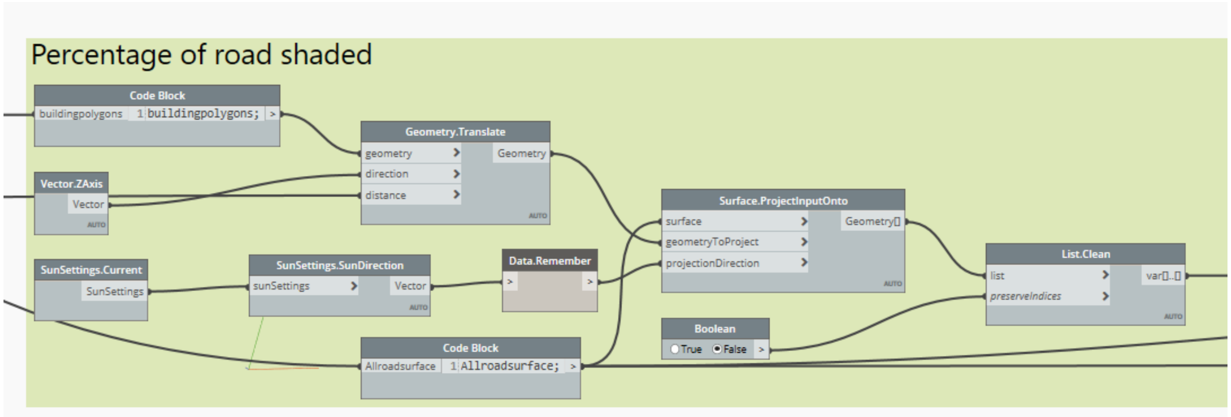


Figure 86: Finding the shading angle and the shadow casting on the road

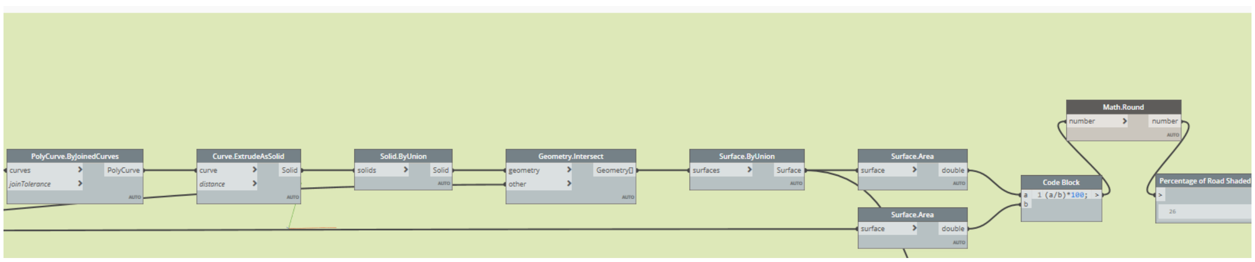


Figure 87: Calculating percentage road area shaded.

Hence, all the six Design Goal parameters have been discussed that help to assess the different design solutions generated by the Generative Algorithm.

1. Profit: Maximizing Number of Plots.
2. Maximizing Percentage Open Space Area.
3. Minimizing Road Area Percentage.
4. Maximizing proximity to school.
5. Maximizing proximity to open spaces.
6. Maximizing percentage of road shaded.

4.4 Optimization study – Generative Design Study Types

The Dynamo graph was used in conjunction with generative design tool, to carry out the optimization research. The parametric model's design variables are used as inputs in the optimization process. These parameters are docked by the generative design tool, which modifies their values.

It keeps track of the results and may thus figure out what input values result in a good design. NSGA-II is used by Dynamo to optimize the designs. The user determines the number of generations and population size.

Dynamo's Generative Design tool provides three more design generation approaches in addition to optimization. However, they are essentially deterministic approaches and do not provide any intelligence. (Rohrmann, 2019)

Randomize

Generates a random configuration of parameters based on user defined number.

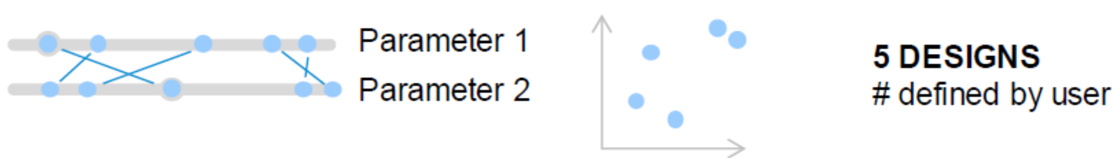


Figure 88: Randomize Study method

Cross Product

Generates all possible design combinations of the parameters to a user defined density.

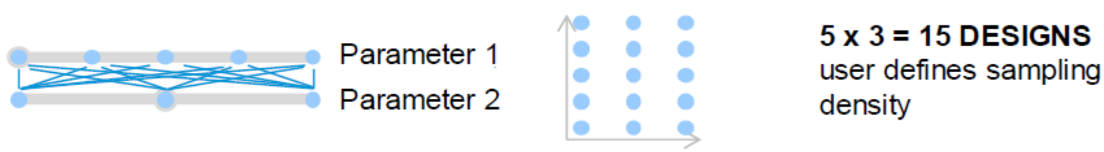


Figure 89: Cross Product Method

Like This

Generates design solution similar to a parameter configuration.

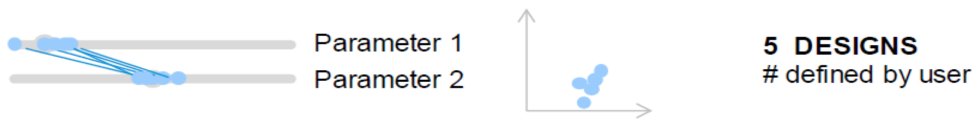


Figure 90: Like this Method

This page is intentionally left blank.

5. Implementation

Two software tools are utilized to carry out the generative design study: Dynamo, a visual programming package, and Generative Design Tool, an optimization tool for the AEC sector.

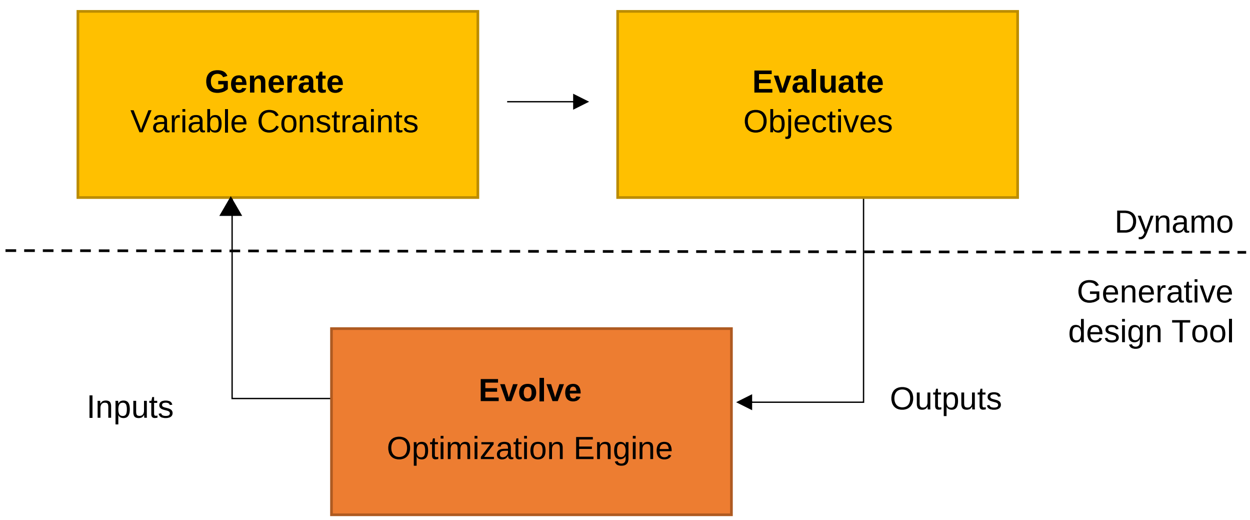


Figure 91: Workflow of the Generative design study

In Dynamo, a parametric model graph is developed that generates a neighborhood layout for the Salt Lake City Neighborhood when paired with the input parameter values. The graph also evaluates the generated design options of the neighborhood and returns their fitness according to the Design Goals.

Generative Design tool is a plugin and uses NSGA-II for its optimization. It is self-contained and has its own user interface. It docks on the graph, generates new sets of input values, and logs the outputs. It discovers which values lead to high-performing solutions and optimizes designs in that direction via crossover and mutation.

5.1 Generate

The generation of design options by the generative algorithm follows a workflow, which is described in the image below. The yellow color represents the constraints, the green color signifies the variables, and the grey color signifies the command.

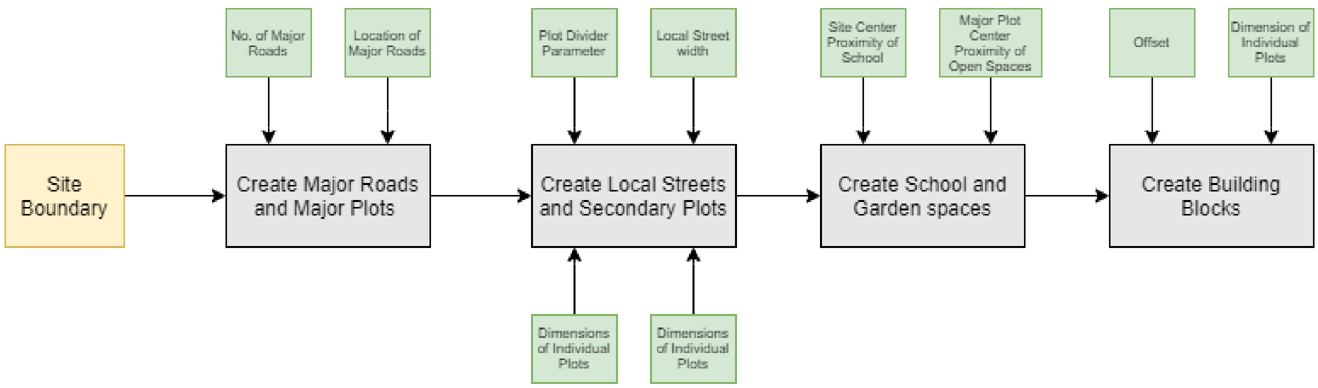


Figure 92: Workflow of the Generative Design algorithm

The site boundary is input by the user, for which the first command is executed which creates the major roads. The major roads are created including three variable parameters of Road width, number of roads and Location of roads.

After the Major roads are created, the genetic algorithm creates the local streets, including three input parameters of Street width, plot divider, dimensions of individual plots.

The individual secondary plots are now assessed based on the centrality measures and garden and school facilities are located on the site.

Finally, the algorithm generates the building blocks based on the offset and block dimension parameter input by the user.

The process of generate is continuously repeated by the Generative algorithm for the different combinations of the input parameters.

5.2 Evaluation of Generated Design Solutions

After the design is generated, its fitness is evaluated based on the six Design Goal indicators. The below section defines the logic behind the scoring criteria of different design goals.

5.2.1 Normalized Metrics

A set of Normalized metrics has been developed for each design goal, to get the average scoring values for each design solution. The scoring is from 0 to 1. 1 being the highest value and the highest desired outcome of each design.

1. Profit: Maximizing Number of Plots.

The Generative Algorithm finds the number of Plots or building generated in the design outcomes. To identify the solution which contains the maximum number of Plots, a normalized metric has been developed to act as a score for each design option.

<u>Normalized Metric for No. of Plots = No, of Plots/1000</u>

For example: The number of plots for the below design option is 612.

Figure 93: Normalized metric for No. of plots

The normalized metric for the No. of Plots would be <u>612/1000 = 0.612</u>

2. Maximizing Percentage Open Space Area Percentage

The open space area is calculated in the algorithm by finding the surface area of the Open space plots. The total open space area is then divided by total site area and multiplied by 100 to get the percentage of open space.

Open area percentage = Total open space area/Site Area * 100

<u>Normalized metric for open space percentage = (Open Space Area Percent * 2) / 100</u>

For example, the open space area percentage of the below design option is 16%.

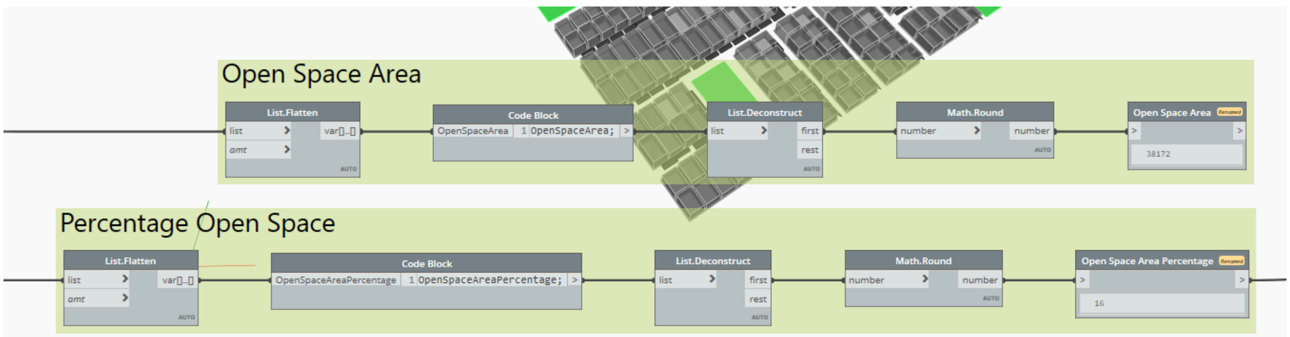


Figure 94: Normalized metric for Open space area %

The normalized metric for the Open space area percentage would be = <u>(16*2)/100 = 0.32</u>

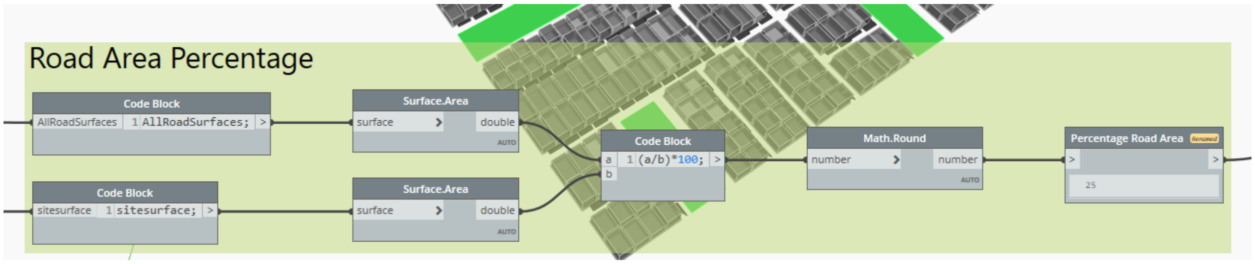
3. Minimizing Road Area Percentage

The road area is calculated by the algorithm, the total road area is then divided by the site are and multiplied by 100 to get the total road area percentage. The normalized metric for the road area is as follows:

Road area percentage = total road area/total site area * 100

<u>Normalized metric for road area percentage = (1- (total road area percent * 2) /100)</u>

For example, in the below design, the total road area percentage is 25.



The normalized metric would be (1 - <u>(25*2)/100) = 0.5</u>

On a scale of 0 to 1, 1 is the least desirable outcome for the design solution.

For a design 2, with road percent 12,

The normalized metric would be = (1- (12*2)/100) = 1 – 3.6 = 6.4

Hence, the normalized metric ensures that the minimal percentage of road has higher score.

4. Maximizing proximity to school

The algorithm calculates the radial distance of school to the individual plots and identifies the longest distance and returns its value.

The normalized metric for the Proximity to school is as follows:

<u>Normalized Metric for Proximity to school = (Longest radial distance / 1000)</u>

For example, in the below design, the maximum distance to school is 469m

The Normalized metric for the Proximity to school = 1 – (469/1000) = (1 - 4.69) = <u>5.41</u>

For a design 2 with Maximal radial distance to school 700 m,

The normalized metric would be 1 – (700/1000) = 1 – 0.7 = 0.3.

Hence design 1 has shorter distance and has higher normalized score.
That is 0.3 < 0.469 and design 1 is more desirable.

5. Maximizing proximity to open spaces

The algorithm calculates the radial distance of individual plots to the open spaces and identifies the longest distance and returns its value.

The normalized metric for the Proximity to school is as follows:
<u>Normalized Metric for Proximity to school = 1 - (Longest radial distance * 2 / 1000)</u>

For example, in the below design, the maximum distance to school is 189m

The normalized metric would be = 1 – (189 * 2)/1000 = 1 – 0.378 = 0.622

For a Design 2, the maximal radial distance to open space is 300m,
The normalized metric would be = 1 – (300 * 2)/1000 = 1 – 0.6 = 0.4

Hence, for the design 1, the radial distance is shorter, then the normalized metric is high. Hence, design 1 is more desirable.

6. Maximizing percentage of road shaded

The shadow caused by the buildings on the road surface is calculate by the algorithm for one specific day. The area covered by shadow is calculated and is divided by the total road area to find the shading percentage.

The normalized metric for the Shading Percentage is as follows:

<u>Normalized Metric for Shading Percentage: (Total Shading Percentage * 2) / 100</u>

For example, for the design below the shading percent is 26 %

The normalized metric for the Shading would be (26 * 2)/ 100 = 0.52

5.2.2 Analysis Graph

For the different Normalized metric values, Spider Graph has been created that assesses the generated design options. Summarizing, the normalized metrics for each design goals are calculated as follows:

1. Normalized Metric for No. of Plots = No, of Plots/1000
2. Normalized metric for open space percentage= (Open Space Area Percent * 2) / 100
3. Normalized metric for road area percentage = (1- (total road area percent * 2) /100)
4. Normalized Metric for Proximity to school = (Longest radial distance / 1000)
5. Normalized Metric for Proximity to school = 1 - (Longest radial distance * 2 / 1000)
6. Normalized Metric for Shading Percentage: (Total Shading Percentage * 2) / 100

For a design option, the normalized metrics are represented in form of Radar chart as shown below:

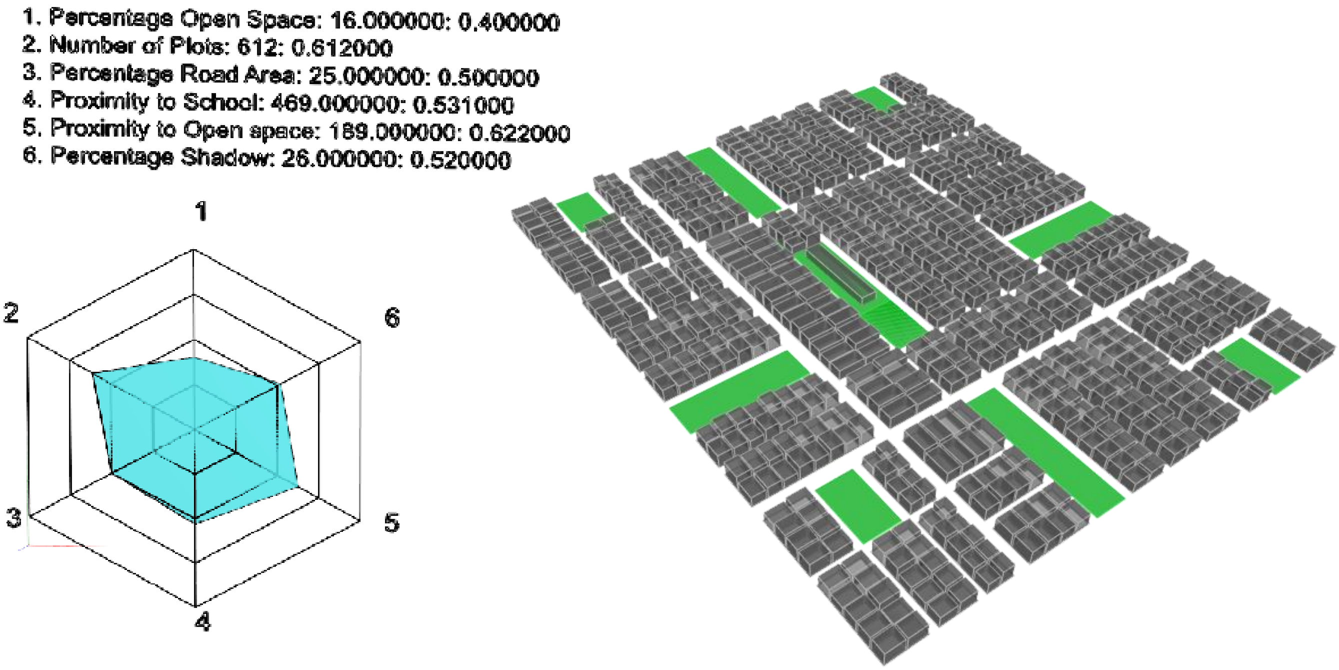


Figure 95: Spider Graph Representation for generated outcomes.

5.3 Presentation of Results

The Generative Design Tool in the Autodesk Revit has been used to generate the design outputs. The following are the Variables and Foals used for the study:

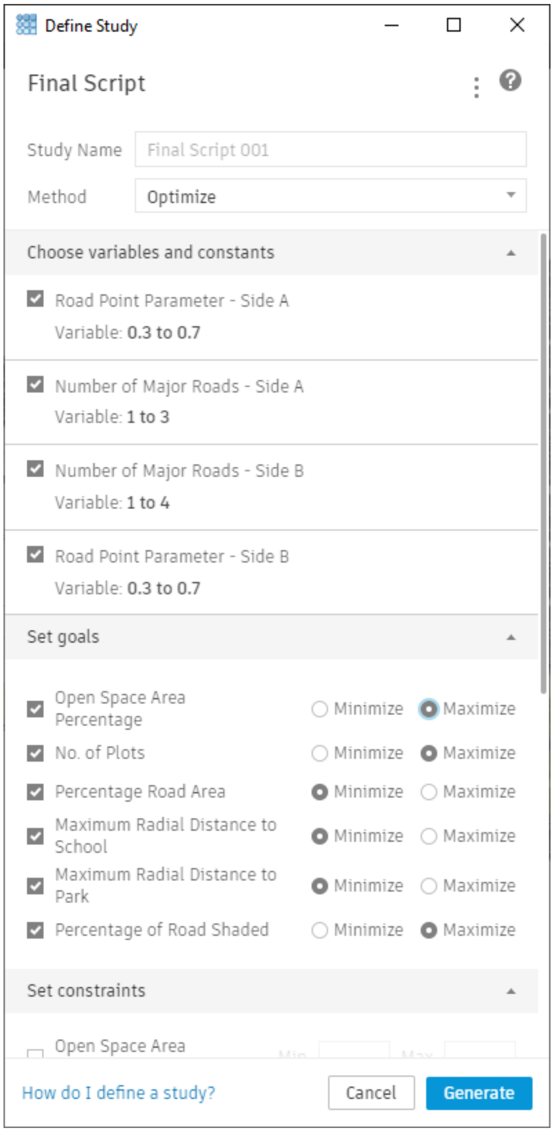


Figure 96: Generative Design study parameters

The Generative Design algorithm has been set to optimize the variables and find 20 iterations of design populations of each population size of 40 outputs.

Total Solutions created = 20 x 40 = 800.

Among the 800 solutions, the Generative algorithm finds the top performing design options and selects the top 60 best performing solutions.

The following are the design solutions created by the algorithm:

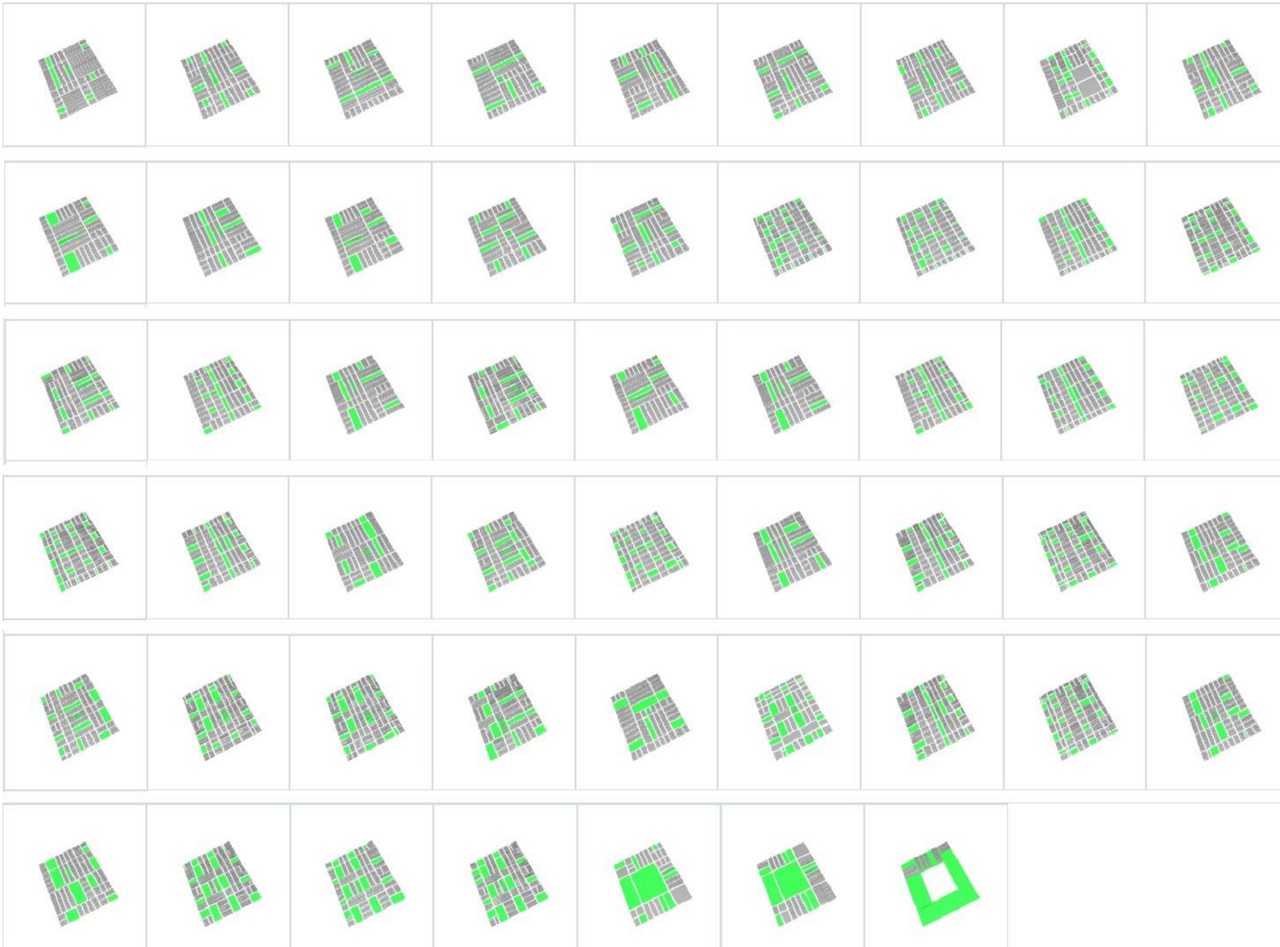


Figure 97: Generated Design options. From top to last - Ascending order of open space size.

The generative design tool assesses the generated design options in the form of visual data representation that helps to visualize the entire design space in the form of Graphs and plots.

Two types of plots are used to represent the variables of the design solutions.

a. Scatter Plot

The scatter plot presents the entire design space in the form of four variables, which are represented on the X, Y axis, Size and Color variations.

The scatter plot for the Optimization study is shown below :

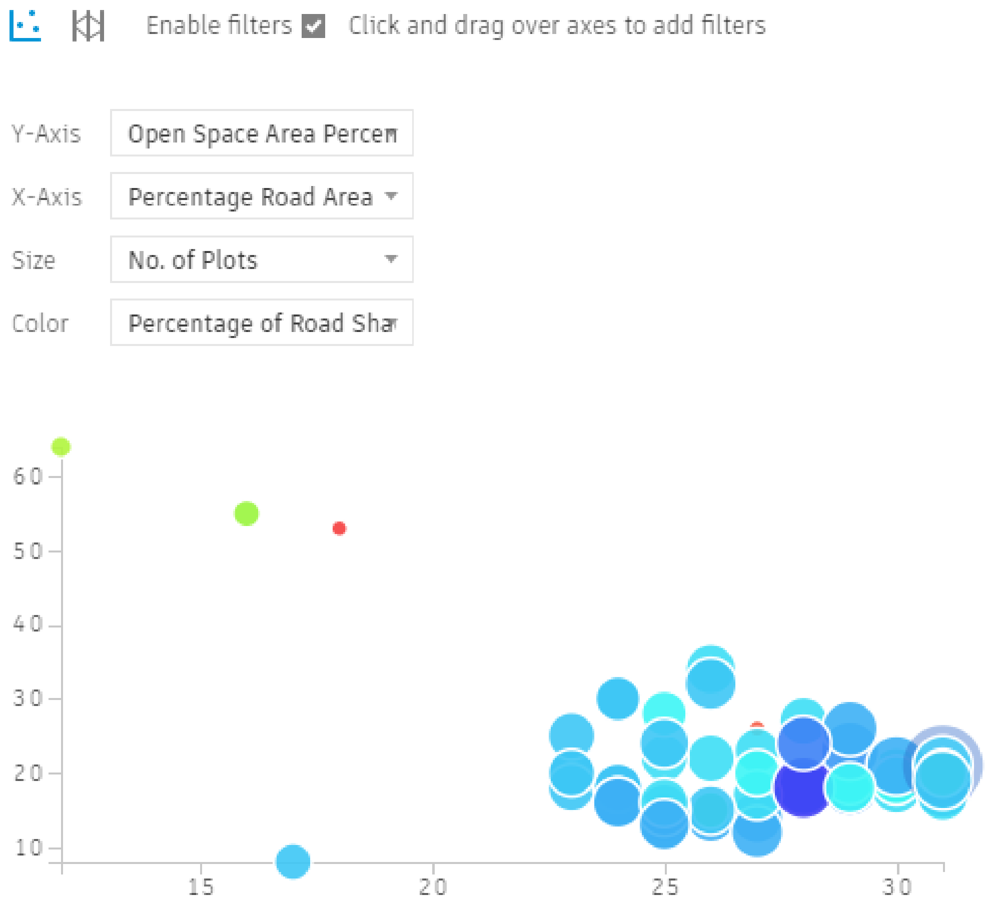


Figure 98: Scatter Plot

b. Parallel Coordinates

The second for of data representation technique used is the Parallel coordinates plot. This visualization includes all the different Variable parameter values and the outcome of the goals.

The Parallel coordinates plot is highly useful to view values of all the parameters of the design solutions.

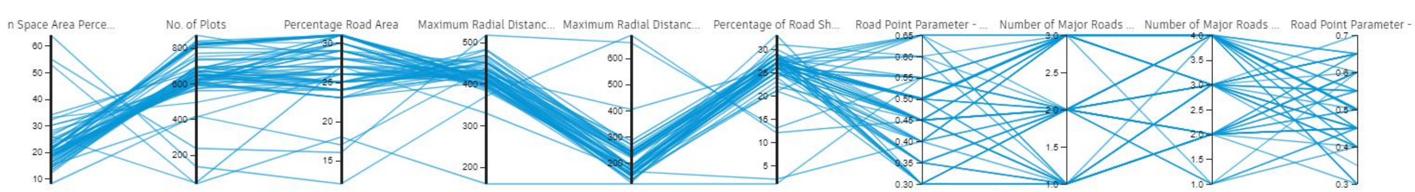


Figure 99: Parallel Coordinates Plot

5.4 Results

This section discusses about the different results drawn from the Generative design study. The results can be divided into to parts: a. Highest Performing specific to one goal, b. Highest performing on each design goal. (average)

5.4.1 Goal specific highest performing design solutions.

The first set of solutions we select are the high performing options of each design goal.

Goal 1 – Maximizing No. of Plots – Finding Design Solution with highest no. of Plots. - Solution 1

Outputs	
Open Space Area Percentage	12.0
No. of Plots	696.0
Percentage Road Area	27.0
Maximum Radial Distance to School	438.0
Maximum Radial Distance to Park	241.0
Percentage of Road Shaded	28.0
Variables	
Road Point Parameter - Side A	0.500
Number of Major Roads - Side A	2
Number of Major Roads - Side B	2
Road Point Parameter - Side B	0.500

Figure 101: Solution 1 - Outputs and Variables

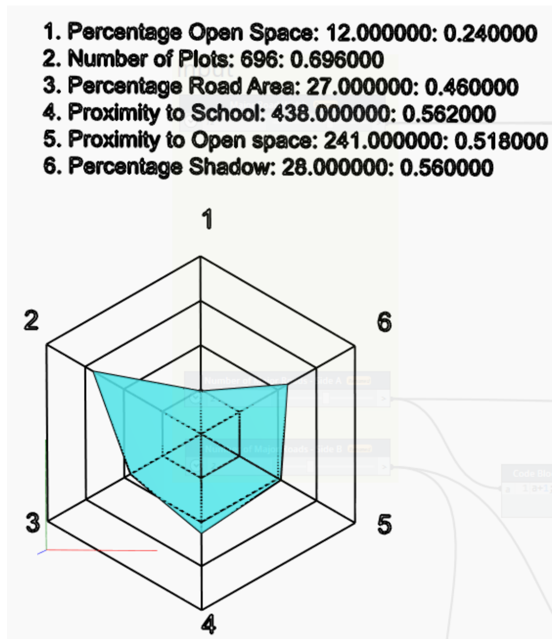


Figure 100: Solution 1 - Radar Chart

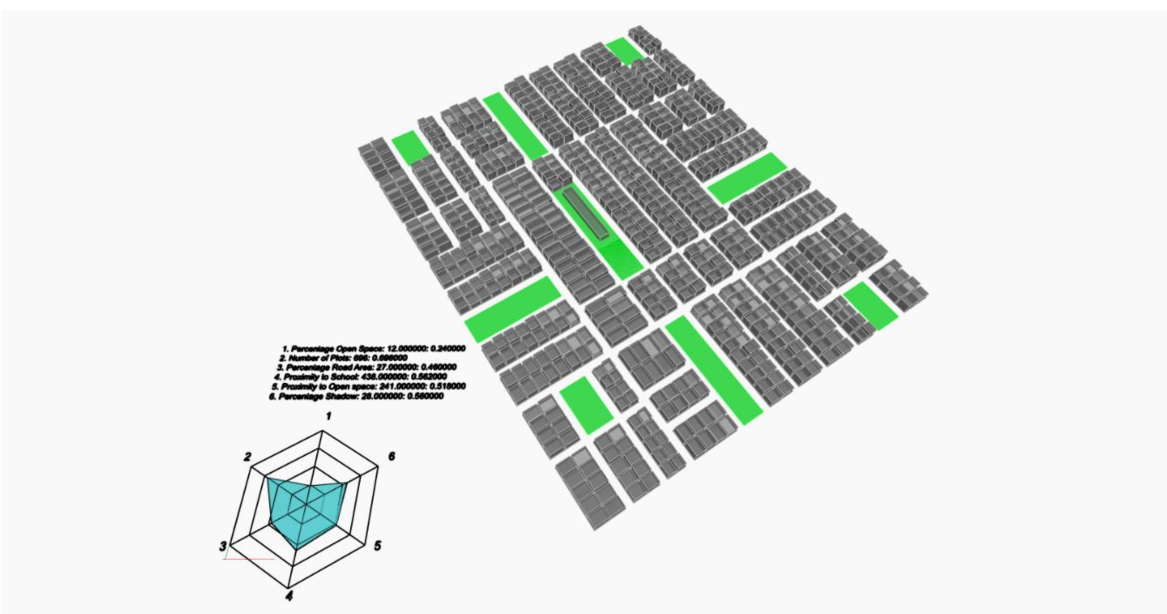
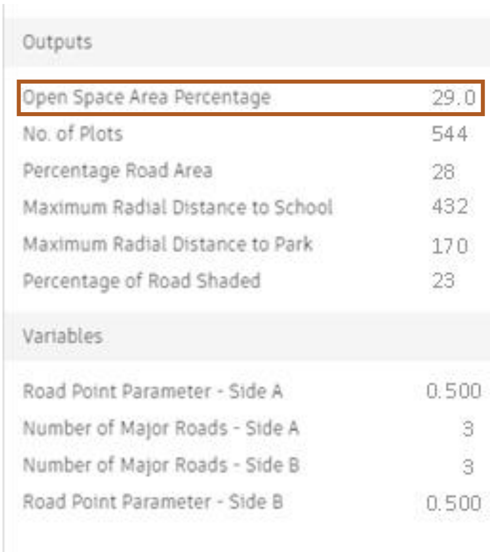


Figure 102: Solution 1 - Design solution with maximum no. of Plots.

Goal 2 – Maximizing Open space area – Finding Design solution with highest open space area percentage. - Solution 2



Outputs	
Open Space Area Percentage	29.0
No. of Plots	544
Percentage Road Area	28
Maximum Radial Distance to School	432
Maximum Radial Distance to Park	170
Percentage of Road Shaded	23
Variables	
Road Point Parameter - Side A	0.500
Number of Major Roads - Side A	3
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.500

Figure 104: Solution 2 - Outputs and Variables

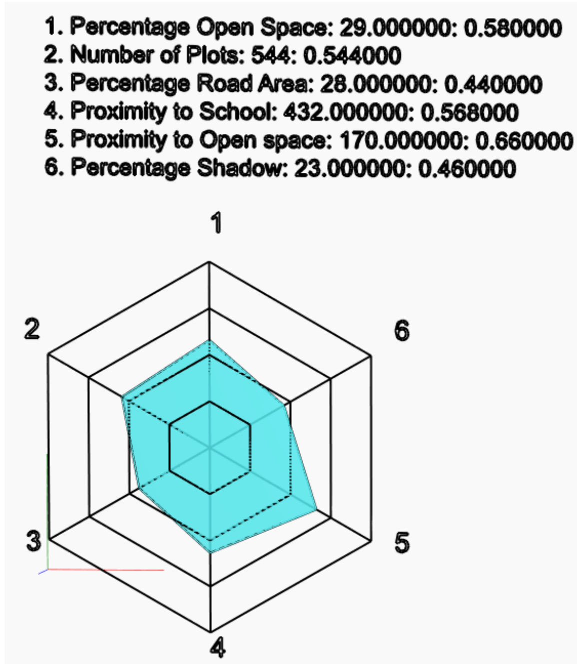


Figure 103: Solution 2- Radar chart

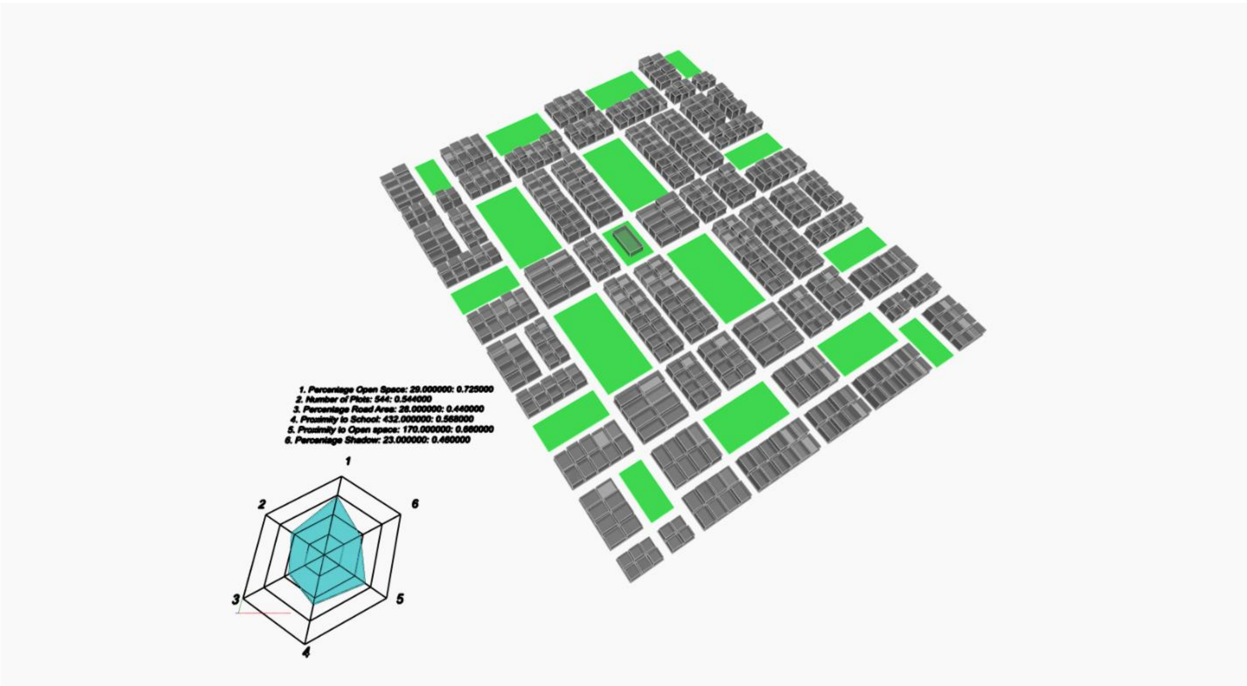


Figure 105: Solution 2 - Maximum Open area %

Goal 3 – Minimize Road Percentage Area – Solution 3

Outputs	
Open Space Area Percentage	20.0
No. of Plots	594.0
Percentage Road Area	23.0
Maximum Radial Distance to School	425.0
Maximum Radial Distance to Park	226.0
Percentage of Road Shaded	27.0
Variables	
Road Point Parameter - Side A	0.650
Number of Major Roads - Side A	1
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.650

Figure 107: Solution 3 - Outputs and Variables

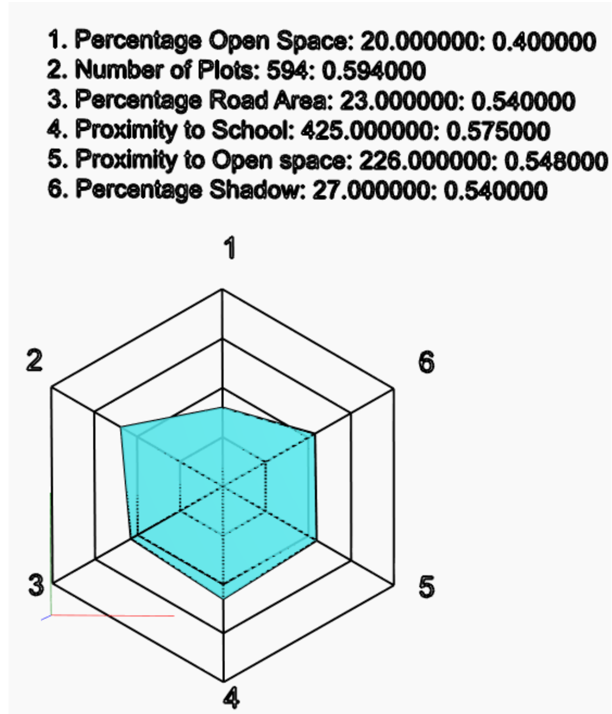


Figure 106: Solution 3 - Radar chart

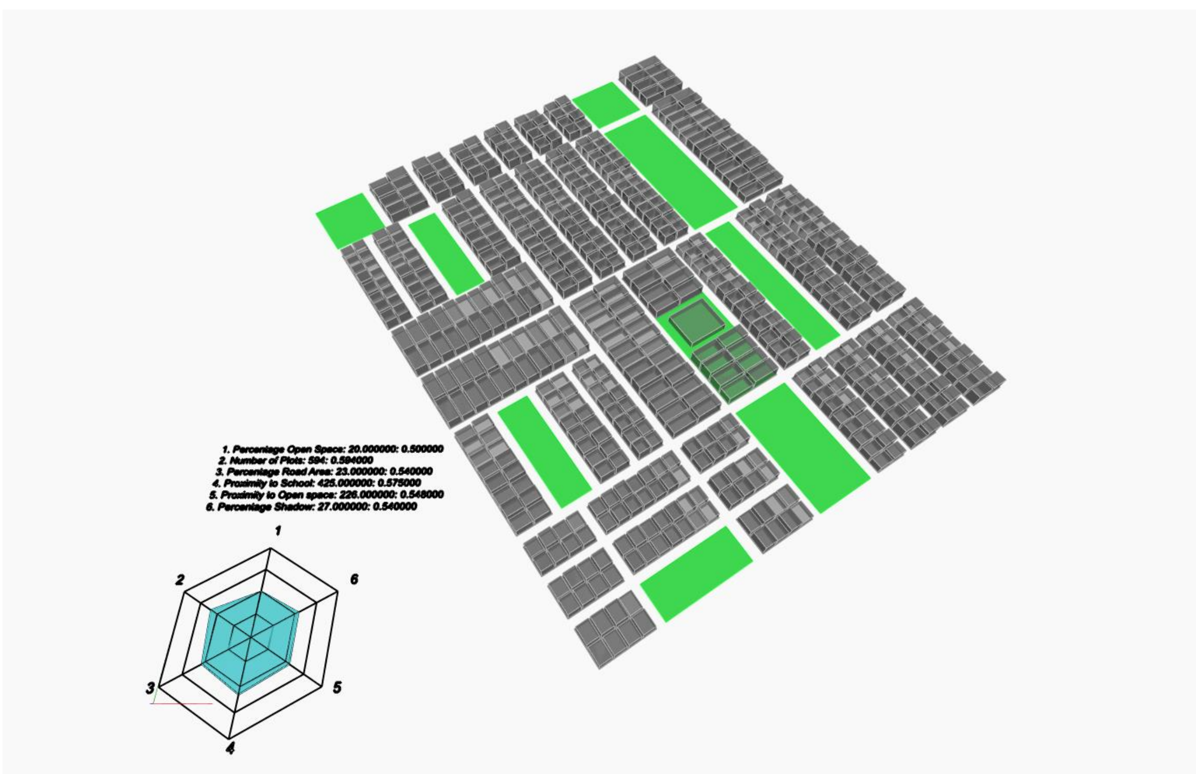


Figure 108: Solution 3 - Minimum Road Area Percentage

Goal 4 – Maximize Proximity to School – Solution 4

Outputs	
Open Space Area Percentage	17.0
No. of Plots	642.0
Percentage Road Area	27.0
Maximum Radial Distance to School	403.0
Maximum Radial Distance to Park	230.0
Percentage of Road Shaded	26.0
Variables	
Road Point Parameter - Side A	0.450
Number of Major Roads - Side A	2
Number of Major Roads - Side B	2
Road Point Parameter - Side B	0.400

Figure 110: Solution 4 - Outputs and Variables

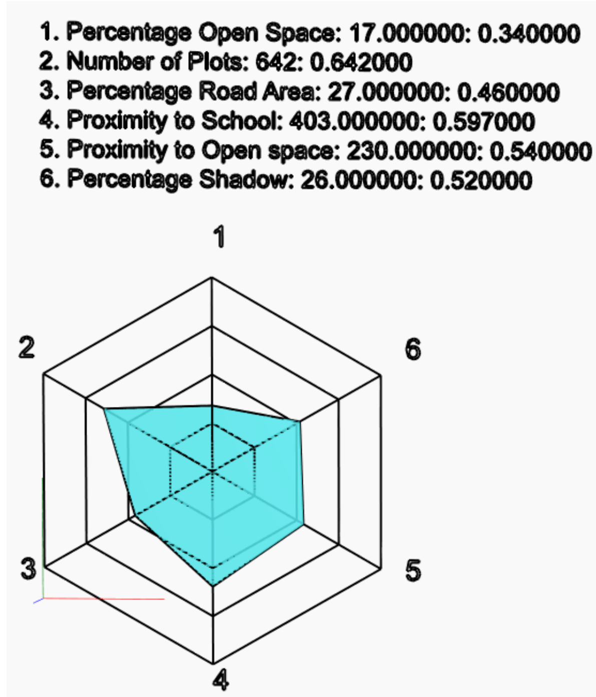


Figure 109: Solution 4 - Radar Chart

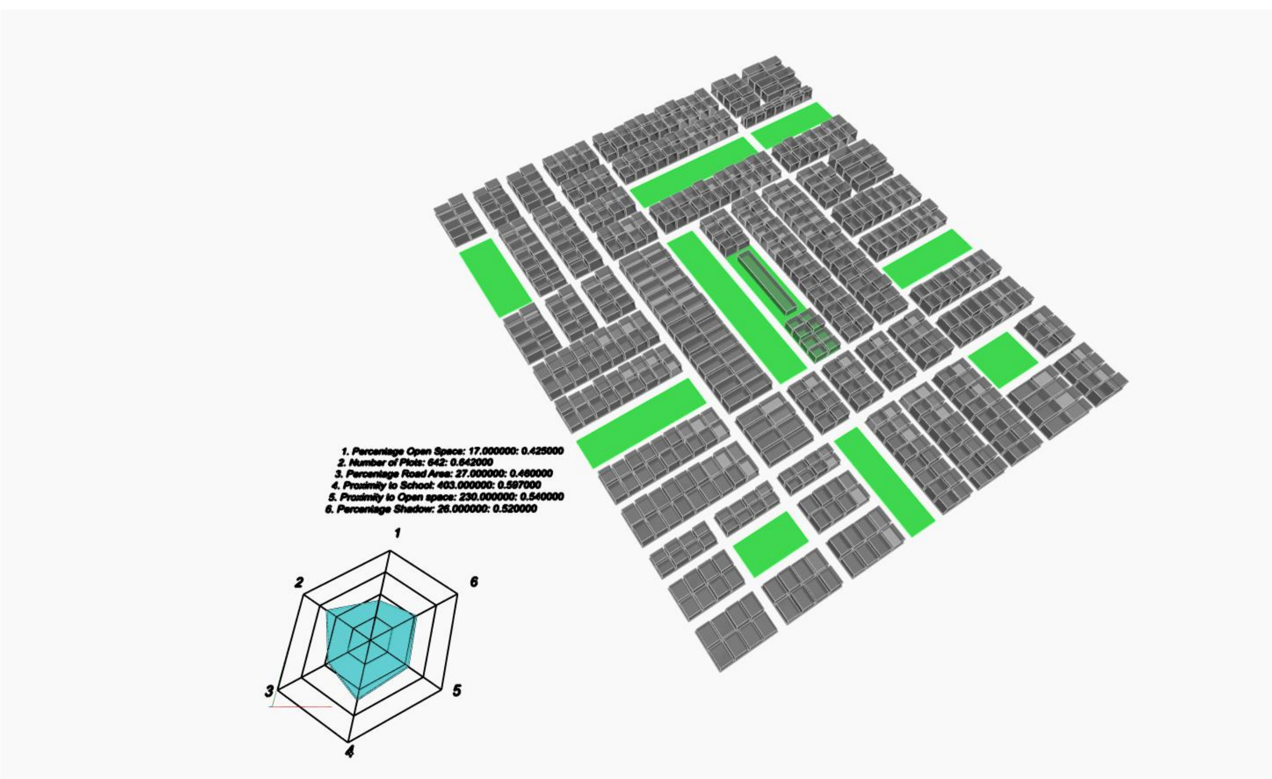


Figure 111: Maximum Proximity to School

Goal 5 – Maximize Proximity to Open Space – Solution 5

Outputs	
Open Space Area Percentage	23.0
No. of Plots	570.0
Percentage Road Area	27.0
Maximum Radial Distance to School	414.0
Maximum Radial Distance to Park	186.0
Percentage of Road Shaded	26.0
Variables	
Road Point Parameter - Side A	0.650
Number of Major Roads - Side A	2
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.500

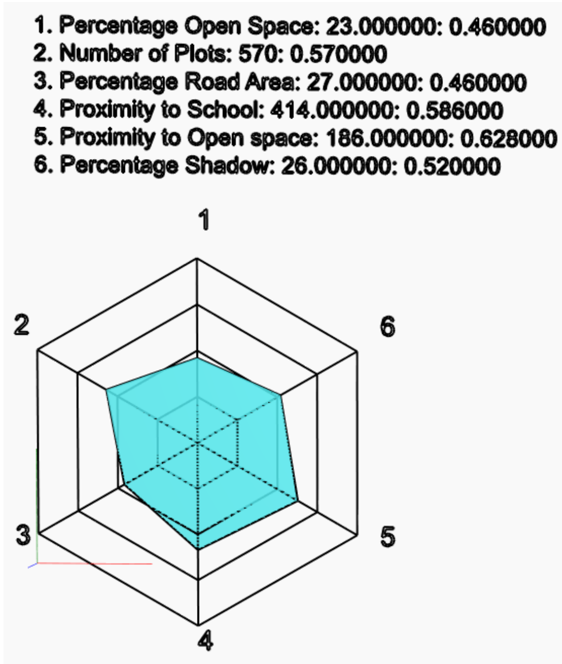


Figure 112: Solution 5 - Outputs and Variables

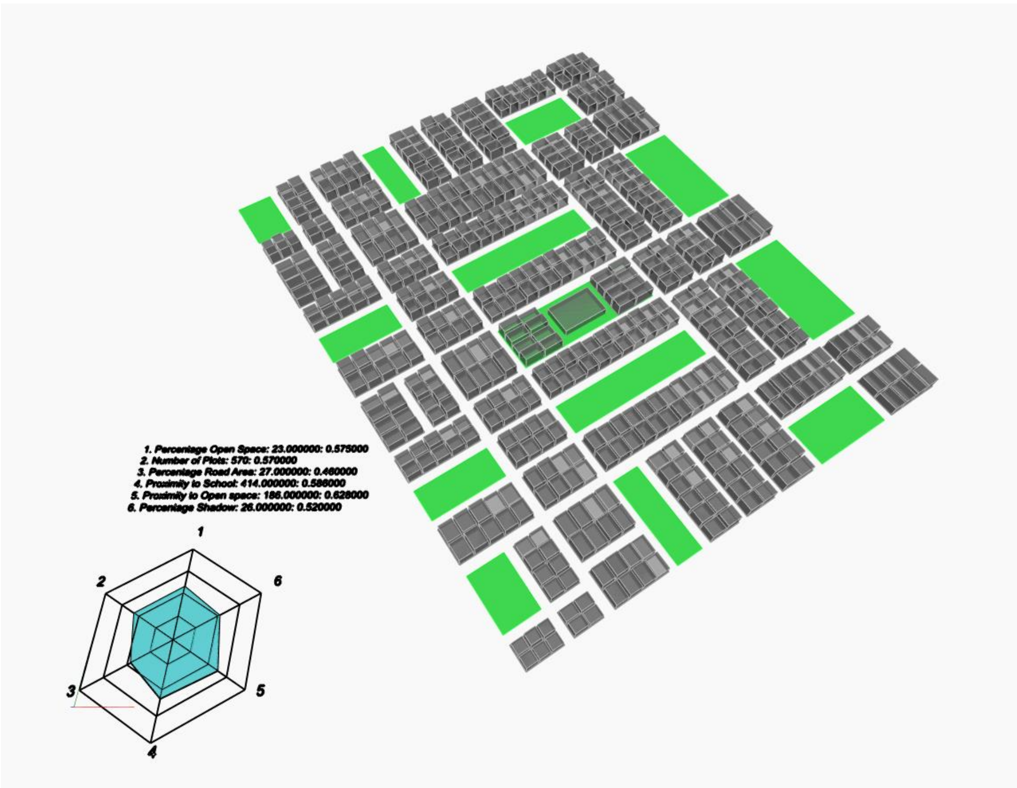


Figure 113: Solution 5 - Maximum Proximity to Open Space

Goal 6 – Maximize Shading Area – Solution 6

Outputs	
Open Space Area Percentage	16.0
No. of Plots	652.0
Percentage Road Area	24.0
Maximum Radial Distance to School	431.0
Maximum Radial Distance to Park	283.0
Percentage of Road Shaded	29.0

Variables	
Road Point Parameter - Side A	0.300
Number of Major Roads - Side A	1
Number of Major Roads - Side B	2
Road Point Parameter - Side B	0.550

Figure 115: Solution 6 - Outputs and Variables

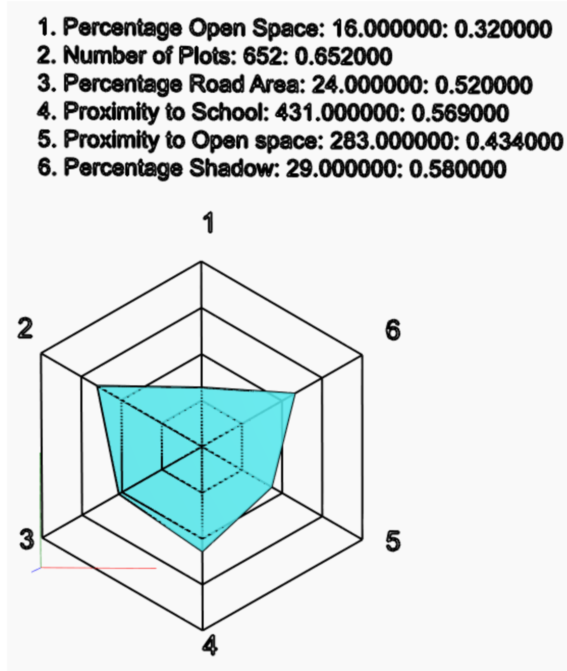


Figure 114: Solution 6: Radar Chart

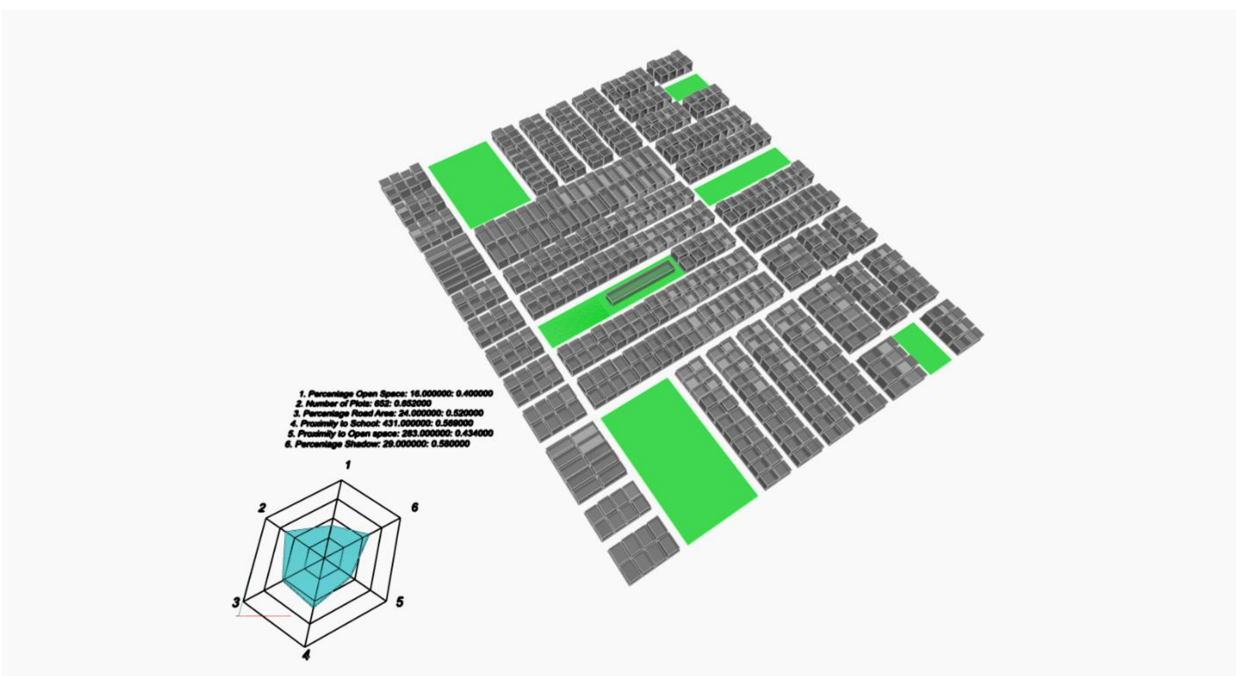


Figure 116: Solution 6 - Maximum Shading Area

5.4.2 Average Performing Solutions:

 # Solution 7:

Outputs	
Open Space Area Percentage	16.0
No. of Plots	628.0
Percentage Road Area	25.0
Maximum Radial Distance to School	428.0
Maximum Radial Distance to Park	189.0
Percentage of Road Shaded	26.0
Variables	
Road Point Parameter - Side A	0.500
Number of Major Roads - Side A	1
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.550

Figure 118: Solution 7 - Outputs and Variables

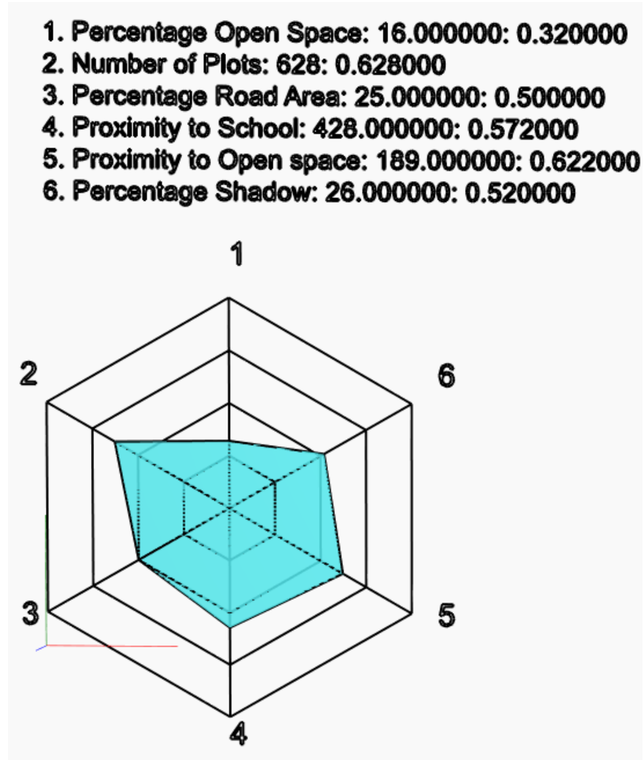


Figure 117: Solution 7 - Radar Chart

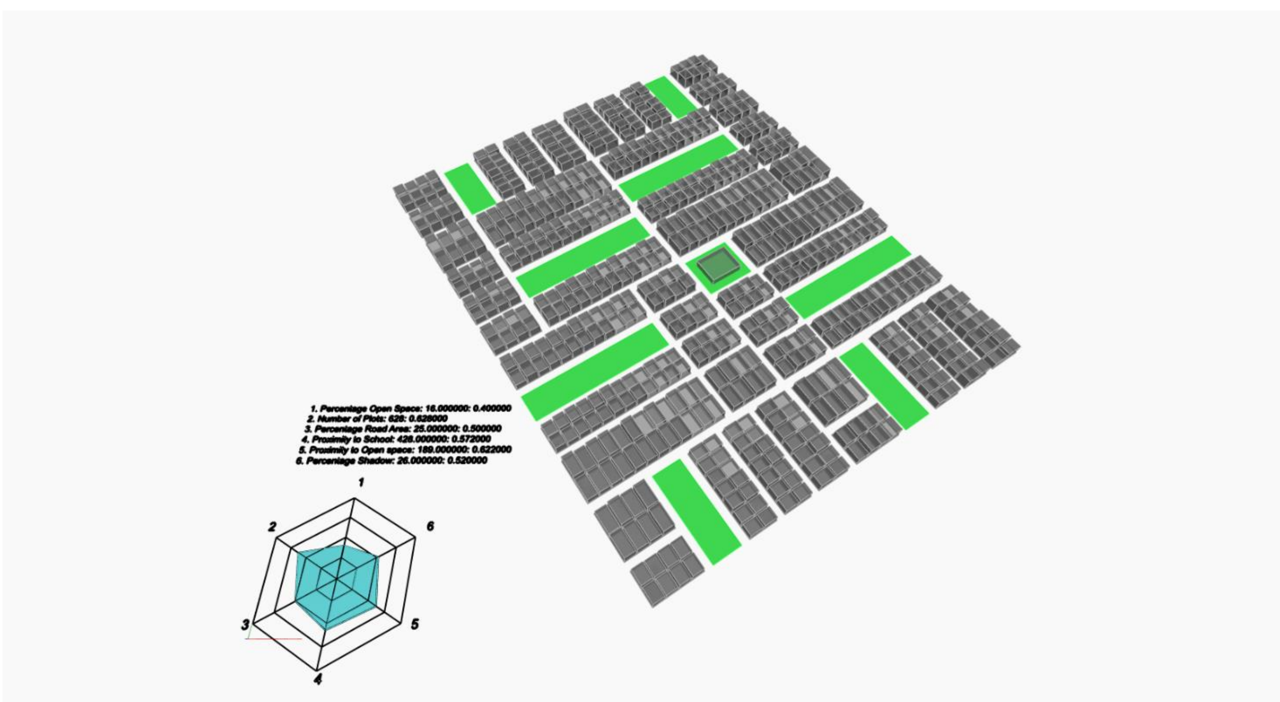


Figure 119: Solution 7 - 3D View

Solution 8

Outputs	
Open Space Area Percentage	20.0
No. of Plots	598
Percentage Road Area	28.0
Maximum Radial Distance to School	435
Maximum Radial Distance to Park	186
Percentage of Road Shaded	25
Variables	
Road Point Parameter - Side A	0.50
Number of Major Roads - Side A	2
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.50

Figure 121: Solution 8 - Outputs and variables

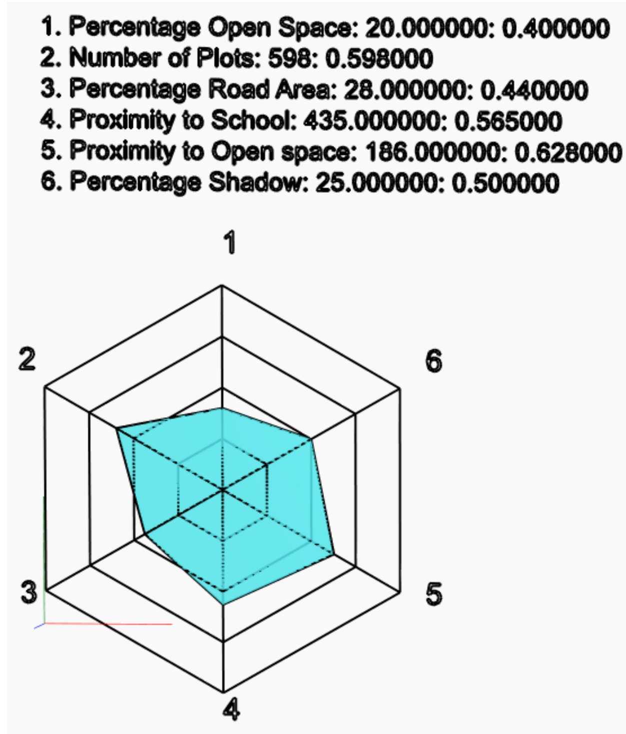


Figure 120: Solution 8 Radar Chart

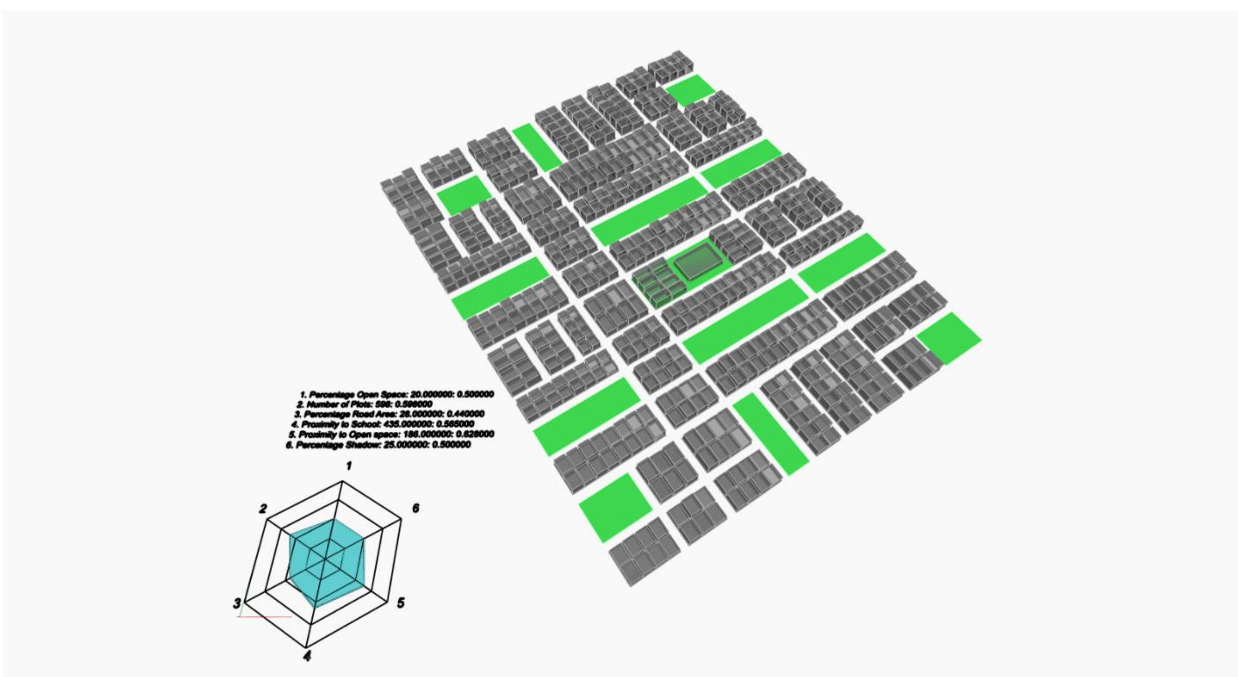


Figure 122: Solution 8 - 3D View

Solution 9

Outputs	
Open Space Area Percentage	16.0
No. of Plots	612
Percentage Road Area	25.0
Maximum Radial Distance to School	469
Maximum Radial Distance to Park	189
Percentage of Road Shaded	26
Variables	
Road Point Parameter - Side A	0.50
Number of Major Roads - Side A	1
Number of Major Roads - Side B	3
Road Point Parameter - Side B	0.50

Figure 124: Solution 9 - Outputs and Variables

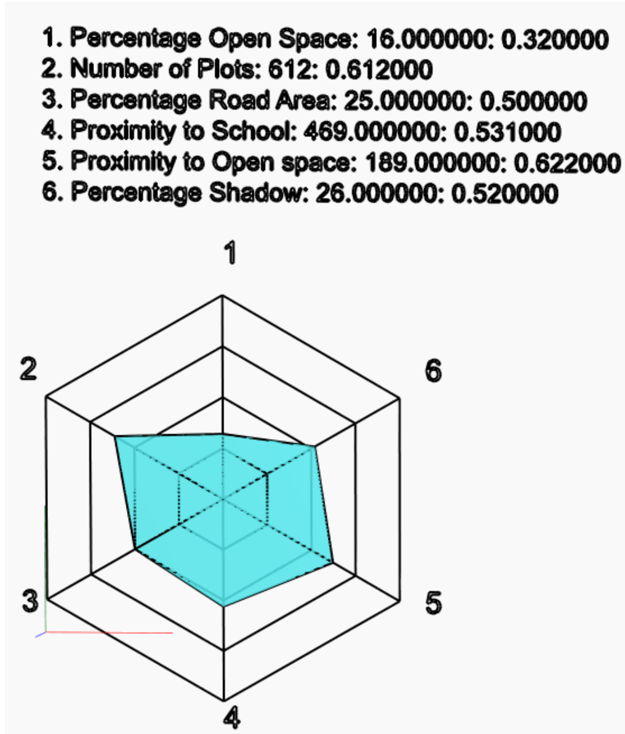


Figure 123: Solution 9 - Radar Chart

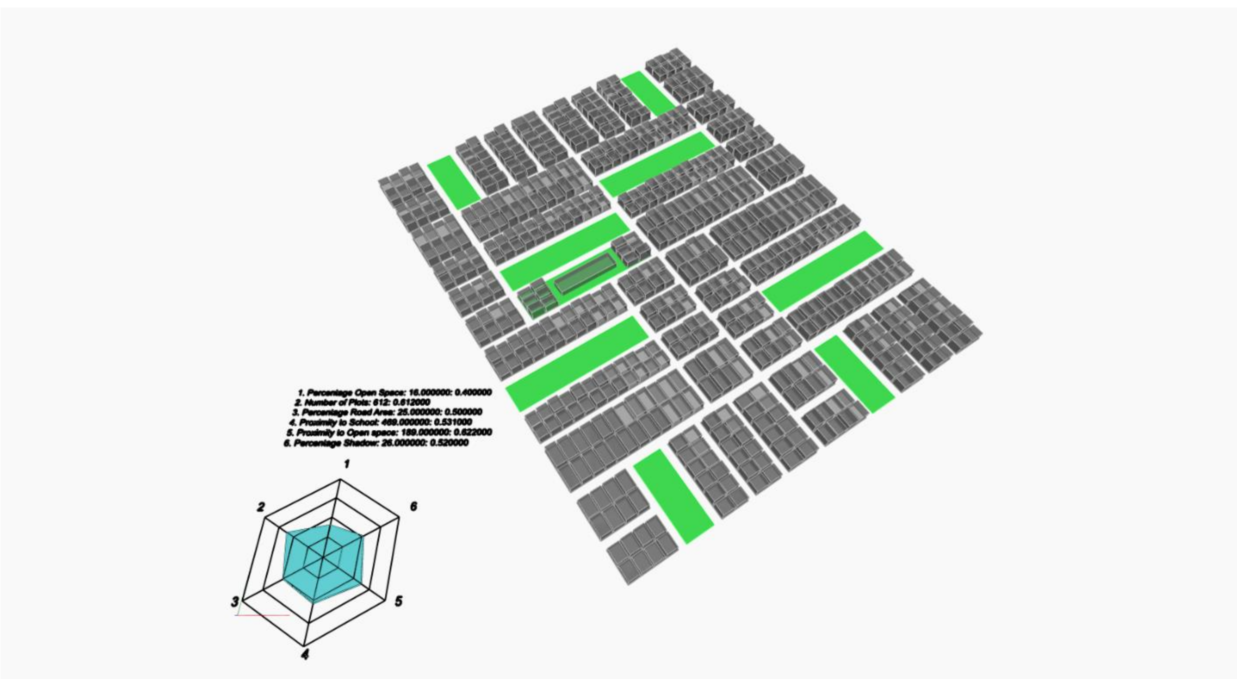


Figure 125: Solution 9 - 3D View

Solution 10

Outputs	
Open Space Area Percentage	15.0
No. of Plots	692.0
Percentage Road Area	25.0
Maximum Radial Distance to School	483.0
Maximum Radial Distance to Park	268.0
Percentage of Road Shaded	28.0
Variables	
Road Point Parameter - Side A	0.650
Number of Major Roads - Side A	1
Number of Major Roads - Side B	2
Road Point Parameter - Side B	0.600

Figure 127: Solution 10 - Outputs and variables

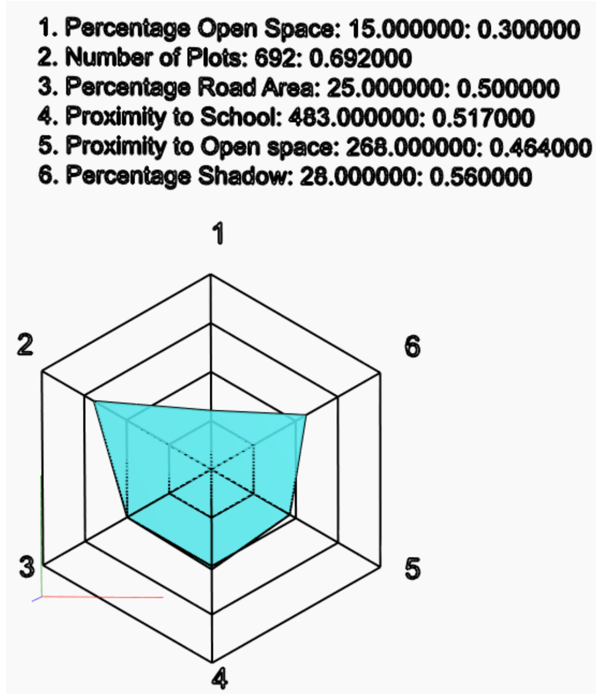


Figure 126: Solution 10 - Radar chart

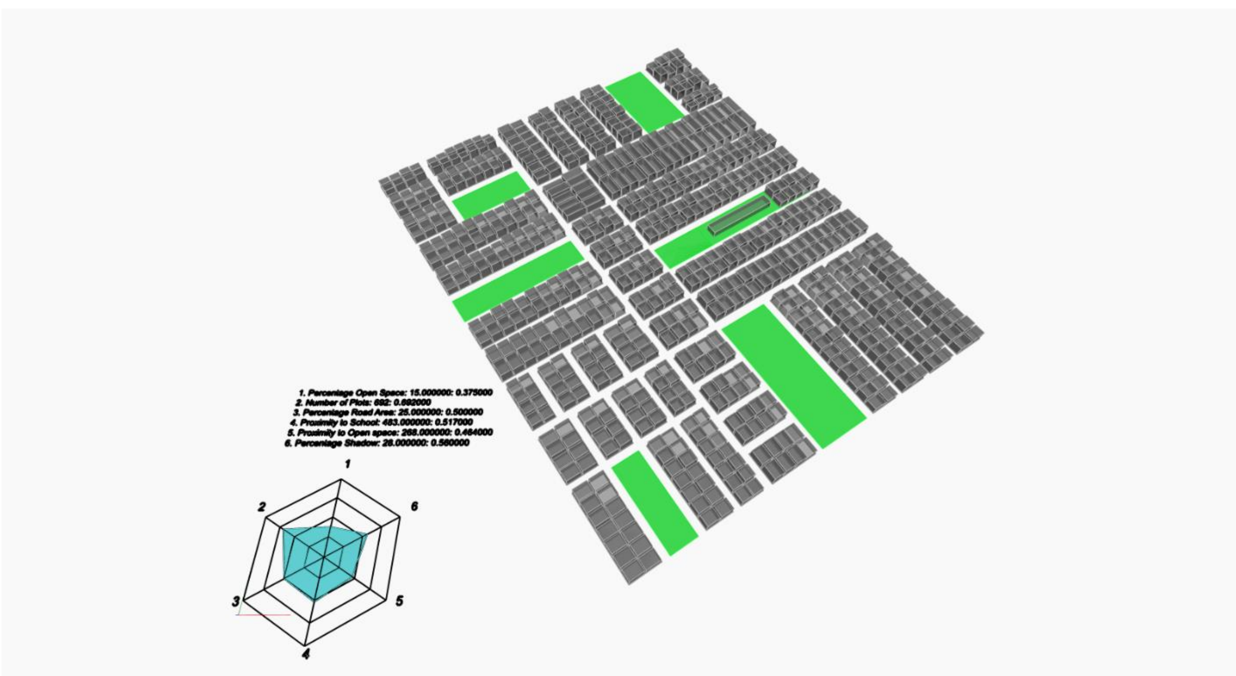


Figure 128: Solution 10 - 3D View

5.4.3 Summary of selected Solutions

The selected 10 outcomes of the Generative Algorithm can be compared with their normalized metrics to find the best average performing solution.

	Normalized Metrics						
	Goal 1 - Max No. of Plots	Goal 2 - Max Open Area %	Goal 3 - Min Road Area %	Goal 4 - Proximity to School	Goal 5 - Proximity to open space	Goal 6 - Max Shading %	Average
Solution 1	0.696	0.300	0.460	0.562	0.518	0.560	0.516
Solution 2	0.544	0.580	0.440	0.568	0.660	0.460	0.542
Solution 3	0.594	0.500	0.540	0.575	0.548	0.540	0.550
Solution 4	0.624	0.425	0.460	0.597	0.540	0.520	0.528
Solution 5	0.570	0.575	0.460	0.586	0.628	0.520	0.557
Solution 6	0.520	0.400	0.520	0.569	0.434	0.580	0.504
Solution 7	0.628	0.400	0.500	0.572	0.622	0.520	0.540
Solution 8	0.598	0.500	0.440	0.565	0.626	0.500	0.538
Solution 9	0.612	0.400	0.500	0.531	0.622	0.520	0.531
Solution 10	0.692	0.375	0.500	0.517	0.464	0.580	0.521

Table 1: Analysis of Generated Design Outcome and finding overall best performing solution.

Hence, the Solution 5 is found to be the best performing solution in aspect to average of all the six Design Goals. The second best performing is Solution 3.

5.5 Comparative Analysis of Existing Design with Generative Design Solutions

#Existing Case

The existing site parameter values include:

a. No. of Plots = 378
b. Site Open Space Area Percentage = 9%
c. Area Covered by Roads = 34%
d. Maximum Radial Distance to school = 592m
e. Maximal radial Distance to Park = 283m
f. Shading Percentage = 17%

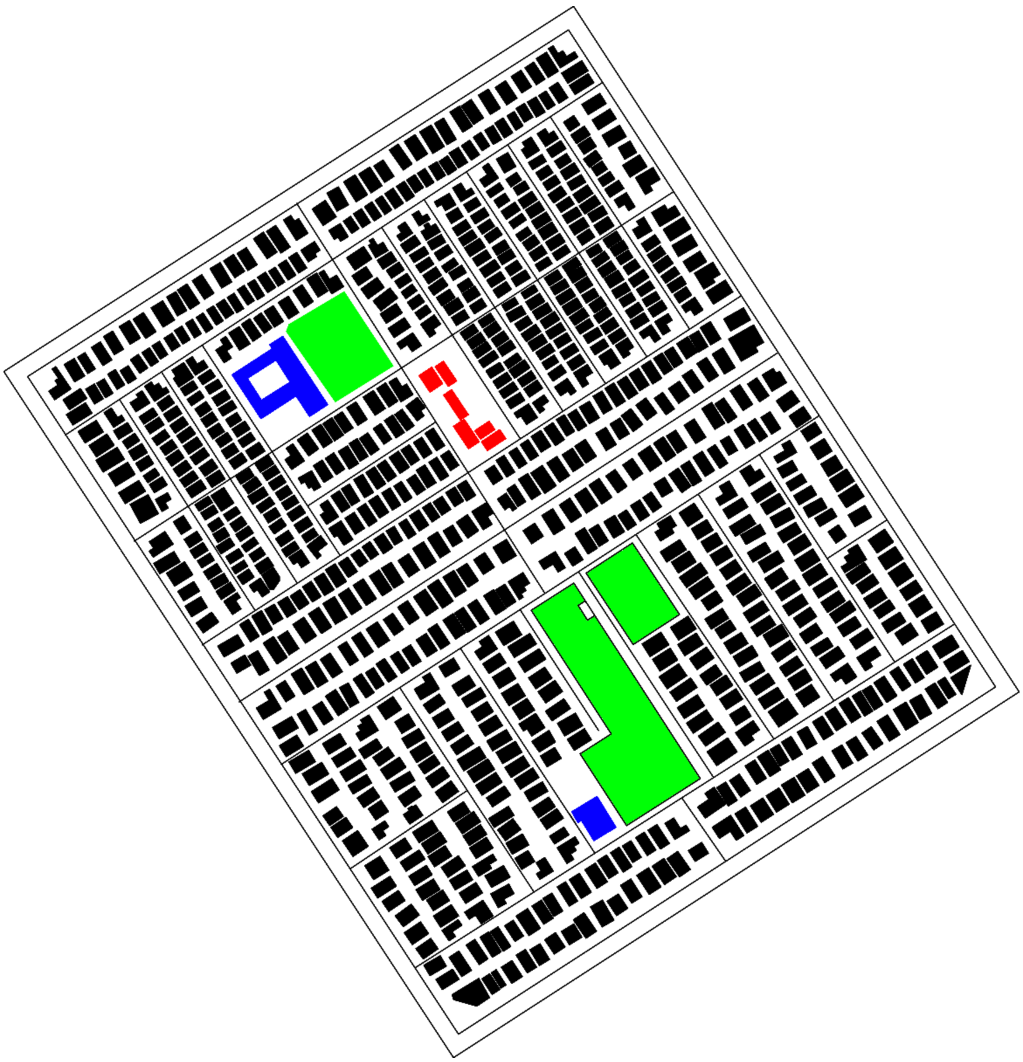


Figure 129: Existing Layout of the BD and CD Block Neighborhood.

#Solution 5

The Solution 5 parameter values include:

a. No. of Plots = 570
b. Site Open Space Area Percentage = 23%
c. Area Covered by Roads = 27%
d. Maximum Radial Distance to school = 414m
e. Maximal radial Distance to Park = 186m
f. Shading Percentage = 26%

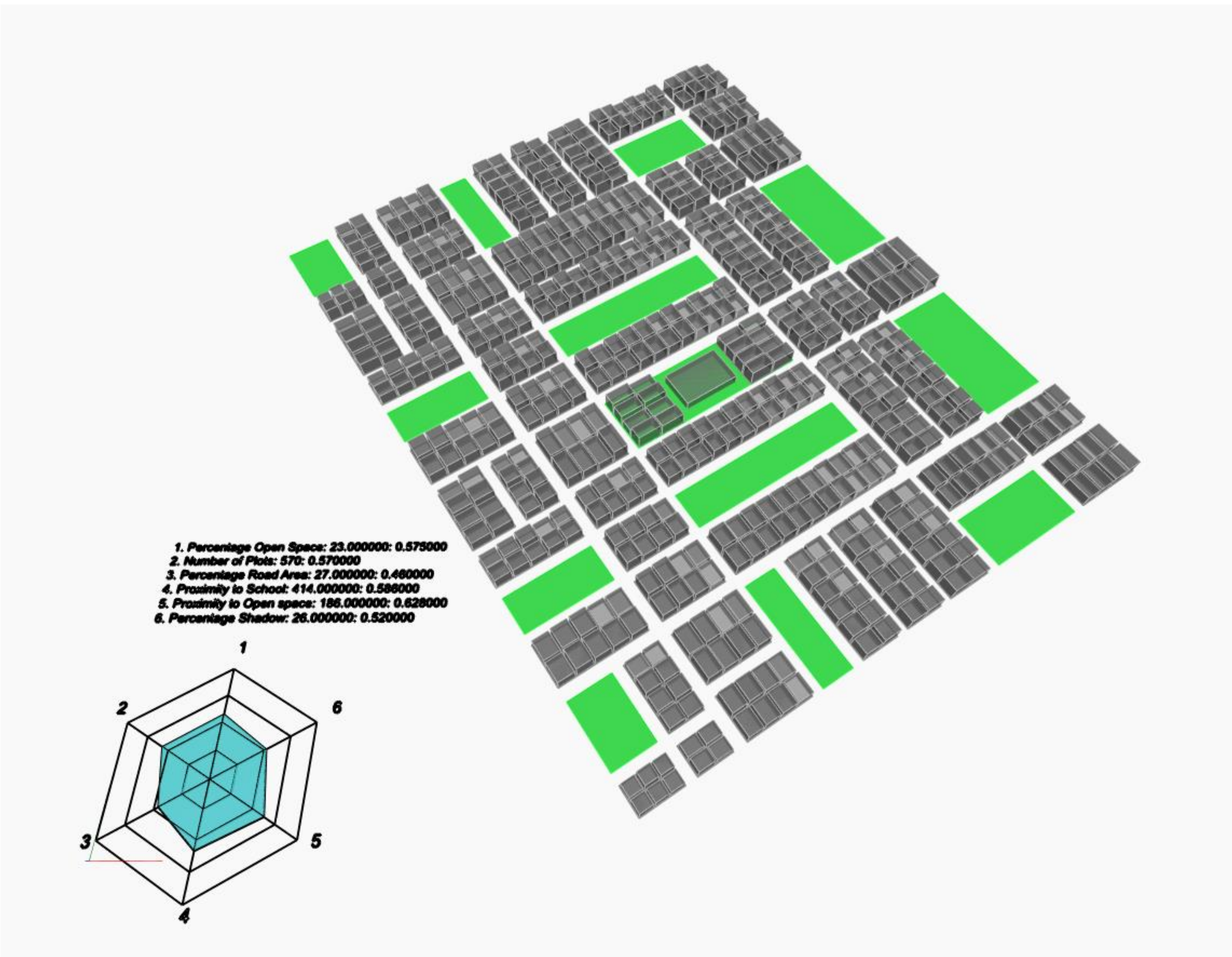


Figure 130: Solution 5

#Solution 3

The Solution 3 parameter values include:

a. No. of Plots = 594
b. Site Open Space Area Percentage = 20%
c. Area Covered by Roads = 23%
d. Maximum Radial Distance to school = 425m
e. Maximal radial Distance to Park = 226m
f. Shading Percentage = 27%

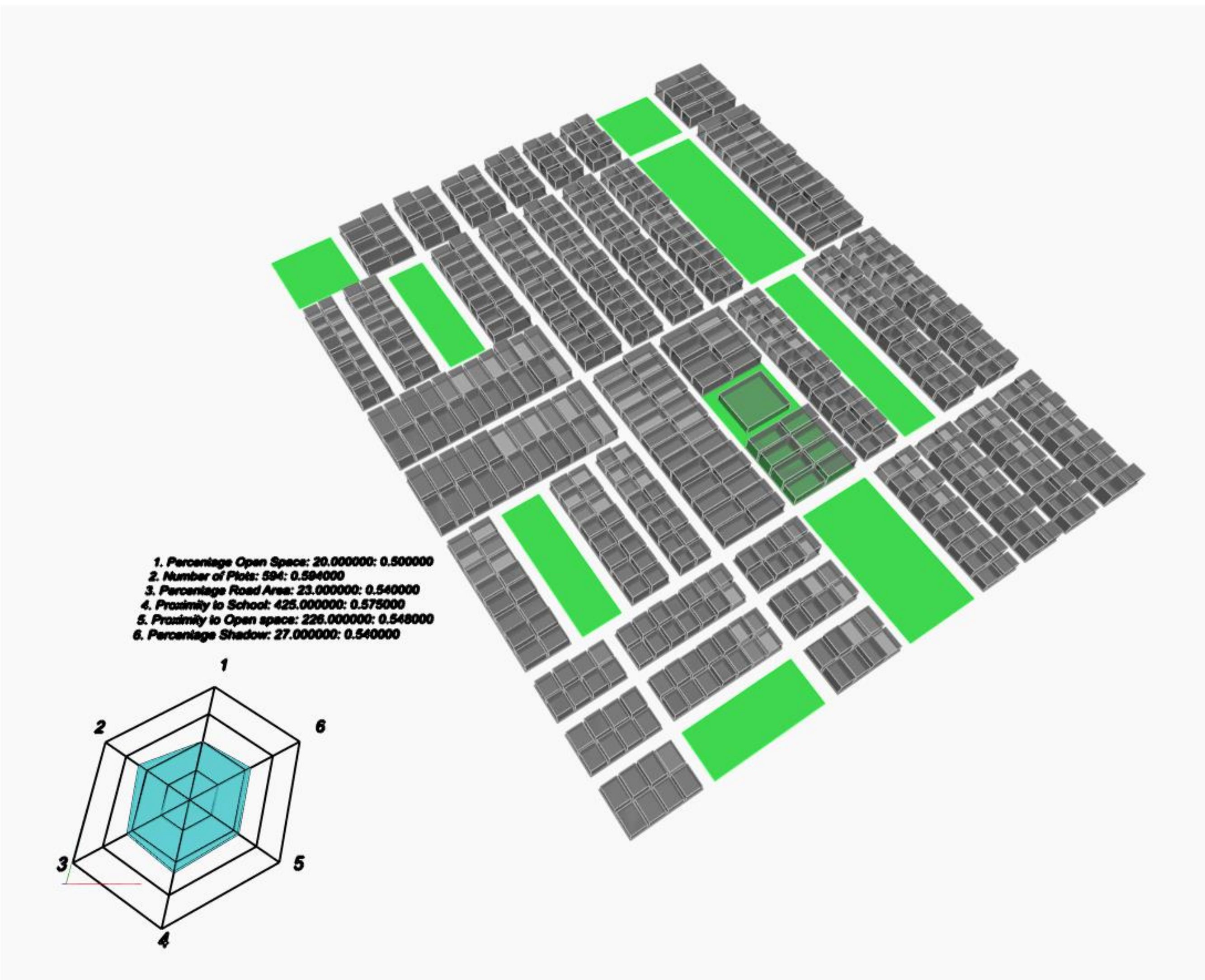
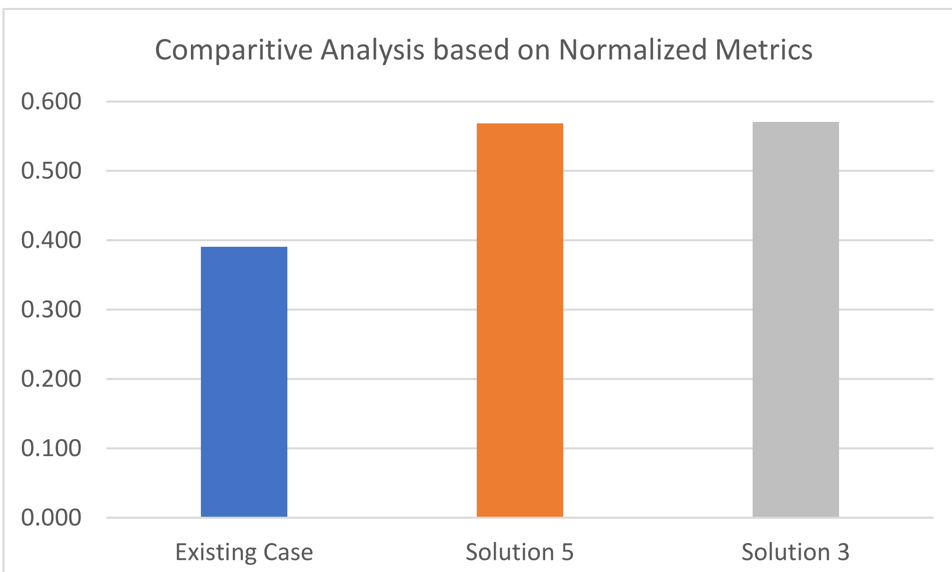


Figure 131: Solution 3

Comparative Analysis

	Output Parameters					
	No. of Plots	Open Space Area %	Total road Area %	Highest Radial Distance to school	Highest Radial Distance to Open Space	Shading %
Existing case	378.000	9.000	34.000	592.000	283.000	17.000
Solution 5	570.000	23.000	27.000	414.000	186.000	26.000
Solution 3	594.000	20.000	23.000	425.000	226.000	27.000

	Normalized Metrics						
	Goal 1 - Max No. of Plots	Goal 2 - Max Open Area %	Goal 3 - Min Road Area %	Goal 4 - Proximity to School	Goal 5 - Proximity to open space	Goal 6 - Max Shading %	Average
Existing Case	0.378	0.180	0.320	0.408	0.717	0.340	0.391
Solution 5	0.570	0.460	0.460	0.586	0.814	0.520	0.568
Solution 3	0.594	0.400	0.540	0.575	0.774	0.540	0.571



Based on the Comparative Analysis, the computer-generated design option is better performing than the Existing design.

This page is intentionally left blank.

6. Conclusion

The Thesis describes the implementation of a Generative Design workflow at a Neighborhood scale in Salt Lake City, Kolkata.

To identify the design success, different design goals were selected which included:

a. Profit – By maximizing no. of Plots and minimizing the number of Roads,
b. Urban quality – By maximizing the Open Spaces
c. Proximity – By minimizing the distances to School and Open Spaces
d. Urban comfort – By maximizing the shading percentage of outdoor spaces.

The project demonstrates how the Generative Design process can create good design outcomes while also doing trade offs between the different conflicting goals of the project. This process ultimately helps to refine the design strategy and find better and more informed final design.

The study generates 800 design options which are evaluated based on certain normalized metrics. These normalized metrics are certain quantifiable measures of the design goals that help to sort the generated design options.

The best performing design solution was found, and the top two solutions were compared to the existing design. Thus, it was found that the computer-generated solutions scored higher than the existing design.

The Impact of the Generative Design in Neighborhood Planning

The global population is increasing at a tremendous rate, this surge in population is also coupled with Urbanization. As more and more people are moving towards cities, agriculture and other use land parcels are now being converted to residential purposes. The increasing pressure on urban areas also emphasizes the magnitude of challenges faced by city planners and architects. The current methods do not adequately accommodate the magnitude of variables that influence the urban fabric. There is a need for robust tools and techniques that would benefit urban planners and architects.

The traditional design approach relies on the intuition and previous experience of the designers which can limit the potential for novel design solutions. Urban design is a complex process that involves multiple parameters such as by-laws and policies, energy optimization, environmental considerations, social factors, design factors, built-uses, proximity, and so on which influence the urban fabric and the quality of living of the inhabitants. Integrating all the different parameters and deriving the optimum design solution is a difficult and complex process. The research provides with a new tool and methodology for Urban design and planning to have better informed and optimized solutions.

In India, cities are expanding at an extremely high pace, with municipal boundaries being expanded, and new agricultural land use change, the Generative Design tool can be used prior to design, so that better informed iterations of the same neighborhood can be found.

Generative Design in an Architect and Planner's friend.

The Generative design tool can be used by the architects and planners to identify the design solutions that are optimized on set of parameters as designed by them. The tool requires architects and planners to select the Project parameters and Design goals, to generate the outcomes.

This tool also allows architects to change the desirability of a goal and find the better performing options. For example, if the Project Goal is to find solutions with maximum Open space area (up to 30%) and maximum shading (up to 50 %), on a later stage if the project goals are to change, say Open space needs to be 40%, and shading 30%, the desirability values of the Generative algorithm can be changes to find the solutions that are now optimized based on the new set goals.

Stakeholder involvement before the Design Process

The Generative Design process allows the architects and planners to take opinions of stakeholders and incorporate them in design, at an early stage of the project. For example, for a neighborhood design, a survey can be done among the stakeholder for their sets of requirements and needs and this survey can be translated into design parameters and goals. These goals can further be ranked based on their importance.

The architects can now input these design variables into the generative algorithm to find the solutions that fit the need of the stakeholders. Then again, the feedback of stakeholders can be taken on the set of generated design solutions.

Changing the set of goals and requirements of the Project is easier in the Generative Design workflow.

The generative design workflow filters the design options based on the Goals of the project. If the project goals are changed at a later design stage, finding new optimum design solution becomes easier through GD.

Future Scope of Work

Although the project has been able to demonstrate the benefits and applicability of the Generative Design process at an Urban Scale, there is a need for further research and testing. The further research is required for more comprehensive Urban design parameters such as climate, safety, comfort, traffic, behavior, etc. These parameters can add more complexity of the Urban design to the generative design and help to create more functional design solutions that incorporate a magnitude of urban design parameters.

This page is intentionally left blank.

7. References

Agarwal, S., & Et. Al. . (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA II. *EEE Transactions On Evulationary Computation, Vol. 6*.

Bardhan, M., & Chatterjee, N. (2016). Urban Morphological Structure of Bindhannagar - A Planned Satellite Township of Kolkata, West Bengal. *Environment and Sustainability* , 264-274.

Braach, M. (2014). Solutions You Cannot Draw. *Architectural Design Volume 84, Issue 5*.

Gabriela Celani, & et. al. (2011). Optimizing the “characteristic structure”. *eCAADe: Respecting Fragile Places*.

Hamid Motieyan, M. S. (2018). An agent-based modeling approach for sustainable urban planning from land use and public transit perspectives. *Cities, Elsevier*.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* The MIT Press.

Hong, S. (2012). Multi Objective Optimization using GA. In Principal Concepts in Applied Evolutionary Computation : Emerging Trends. *IGI Global*.

Janssen, P. (2009). An evolutionary system for design exploration. *Proceedings of the 13th International CAAD Futures Conference*.

Koenig, R., Miao, Y., Aichinger, A., Knecht, K., & Konieva, K. (2020). Integrating urban analysis,generative design, and evolutionary optimization for solving urban design problems. *EPB: Urban Analytics and City Science, SAGE*.

Koenig, Reinhard. (2018). Computational urban design prototyping: Interactive planning synthesis methods—a case study in Cape Town. *International Journal of Architectural Computing, SAGE*.

Koenig, Reinhard. (2020). Integrating Urban Analysis, Generative Design, and Evolutionary Optimization for solving Urban Design Problems. *Urban Analytics and City Science*.

Lawson, B. (2005). *How Designers Think.* London: Routledge.

Miettinen, K. (1998). *Nonlinear Multiobjective Optimization.* New York: Springer US.

Multi Objective Optimization. (2021, 1 10). *Multi-objective optimization*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Multi-objective_optimization#:~:text=Multi%2Dobjective%20optimization%20(also%20known,involving%20more%20than%20one%20objective

Nagy, D. (2017, 1 24). *The problem of learning*. Retrieved from Medium: https://medium.com/generative-design/generative-design-introduction-64fb2db38e1

Nagy, D. (2017a, 1 24). *Learning from nature*. Retrieved from Medium: https://medium.com/generative-design/learning-from-nature-fe5b7290e3de

Nagy, D. (2018). Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout. *Conference: Proceedings of the Symposium on Simulation for Architecture and Urban Design*.

Rohrmann, J. (2019). *Design Optimization in Early Project Stages.* Munich: Technical University of Munich.

Rudi Stouffs, Y. R. (2015). Generative and evolutionary design exploration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*.

Scott, e. (2001). Investigating Human-Computer. *Mitsubishi Electric Research Laboratories*.

Thomas Wortmann, G. N. (2016). Black-box optimisation methods for architectural design. *CAADRIA*.

Turrin, M., & et. al. (n.d.). Design explorations of performance driven geometry in architectural design using parametric modelling and genetic algorithms.

UN. (2020, 2 29). *United Nations - Population Dynamics*. Retrieved from World Population Prospects: population.un.org

Villaggi, L., & Nagy, D. (2018). Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout. *SimAUD*.

Villaggi, Lorenzo. (2017). Project Discover: An Application of Generative Design for Architectural Space Planning. *SimAUD*.

Walmsley, K., & Villaggi, L. (2018). Generative Urban Design. *Sim AUD*.

Wortmann, T., & et. al. . (2015). Advantages of surrogate models for architectural design optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* .