ADAPTABLE NETWORK GENERATOR (ANG):
A GENERATIVE SYSTEM PROPOSAL WITH RESPECT TO TEMPORAL SPACE DESIGN

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BİLGE GÖKTOĞAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTATIONAL DESIGN AND FABRICATION TECHNOLOGIES
IN
ARCHITECTURE

FEBRUARY 2014

Approval of the thesis:

ADAPTABLE NETWORK GENERATOR (ANG):
A GENERATIVE SYSTEM PROPOSAL WITH RESPECT TO TEMPORAL SPACE DESIGN

Submitted by **BİLGE GÖKTOĞAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computational Design and Fabrication Technologies in Architecture in Architecture Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Güven Arif Sargın
Head of Department, **Architecture** ________________

Assoc.Prof. Dr. Arzu Gönenç Sorguç
Supervisor, **Architecture Department, METU** ________________

Examining Committee Members:

Prof. Dr. Ahmet Can Baykan
Architecture Department, METU ________________

Assoc.Prof. Dr. Arzu Gönenç Sorguç
Architecture Department, METU ________________

Assist. Prof. Dr. Başak Uçar Kırmızıgül
Architecture Department, TED University ________________

Assoc. Prof. Dr. Ali Murat Tanyer
Architecture Department, METU ________________

Dr. Mehmet Koray Pekeriçli
Architecture Department, METU ________________

Date: **07.02.2014**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : BİLGE GÖKTOĞAN
Signature :

ABSTRACT

ADAPTABLE NETWORK GENERATOR (ANG):

A GENERATIVE SYSTEM PROPOSAL WITH RESPECT TO TEMPORAL SPACE DESIGN

Göktoğan, Bilge
M.Sc, Computational Design and Fabrication Technologies in Architecture
Supervisor: Assoc. Prof. Dr. Arzu Gönenç Sorguç

February 2014, 95 pages

This thesis is an investigation focusing on generative systems with respect to temporary space organizations. The aim of this study is to discuss the computational model proposed in this thesis and its applications on temporary space design. The research combines computational analysis methods and the generative systems under a multi-layered computational model. In the scope of this thesis two new computational tools based on shortest-path and traveler salesman problem algorithms have been developed. The structure of this computational model and its results are discussed over different cases with respect to temporary space organizations.

Keywords: Generative Systems, Spatial Analysis, Temporary Space Organizations, Network Models, Graph-based Analysis, Computational Design

ÖZ

UYARLANABİLİR AĞ ÜRETİCİSİ (ANG):
GEÇİCİ MEKAN TASARIMLARI İÇİN ÜRETKEN SİSTEM ÖNERİSİ
ADAPTABLE NETWORK GENERATOR (ANG):

Göktoğan, Bilge
Yüksek Lisans, Mimarlıkta Sayısal Tasarım ve Üretim Teknolojileri
Tez Yöneticisi : Doç. Dr. Arzu Gönenç Sorguç

Şubat 2014, 95 Sayfa

Bu tez, üretken sistemlerin geçici mekan tasarımlarına uyagulanmasını konu almaktadır. Bu çalışmanın amacı geçici mekan tasarımları başlığı kapsamında uygulanmak üzere geliştirilen sayısal modelin ve bu modelin uygulamalarının tartışılmasıdır. Yapılan araştırma ve çalışmalar bilgisayar destekli analiz yöntemlerini ve üretken sistemleri çok katmanlı bir model altında birleştirmektedir. Önerilen sayısal model bu tez kapsamında geliştirilmiş iki adet özgün aracı içermektedir. Bu sayısal araçlar en kısa yol ve gezgin satıcı problem algoritmalarını temel almaktadır. Bu sayısal modelin yapısı ve ürettiği sonuçlar geçici mekan organizasyonları kapsamında çeşitli örnek projeler üzerinden tartışılmıştır.

Anahtar Kelimeler: Üretken Sistemler, Mekansak Analiz, Geçici Mekan Tasarımı, Ağ Modelleri, Çizge Temelli Analizler, Sayısal Tasarım

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my supervisor Assoc.Prof. Dr. Arzu Gönenç Sorguç for her guidance, advice, constructive criticisms and most importantly patience throughout my studies.

I am also grateful to my jury members Prof. Dr. Can Baykan, Assoc. Prof. Dr. Arzu Gönenç Sorguç, Assist. Prof. Dr. Başak Uçar Kırmızıgül, Assoc. Prof. Dr. Ali Murat Tanyer, Dr. Mehmet Koray Pekeriçli.

And I especially thank my dearest friends; Aslıhan Günhan, Seray Türkay, Esatcan Coşkun, Ali Yücel Özdemir, Oganalp Canatan, Elif Bilge, Emre Durmaz, Onur Özkoç, Ferhan Ferhat Balık, Emrecan Sevdin, Sezen Dinçel and Onur Özelgül for their limitless love and friendship.

I am also deeply grateful to Müge Kruşa, Kiki Kossmann, Berna Külahçı, Baver Barut, Gökhan Ongun and Cem Korkmaz without whom I would be lost in the streets of Netherlands.

Finally I would like to thank to my family. The greatest woman in my life, my dearest mother İpek Göktoğan and the strongest man I know, my father İbrahim Göktoğan to whom I will be forever indebted. Nothing can be enough to express my gratefulness for their love, support and trust in me during my studies. I also would like to thank to my brother Efe Göktoğan for his sparkling footsteps that helped me find my way whenever I am lost.

Thank you all very much.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. General

Computation has affected the way architectural form, space and structure perceived and realized. It changed the perception of form, the way form is subjected and the way it is produced. Computational theory and techniques have a wide range of concepts. Although form is one of the old and primary topics, it is a minor component of environment, and environment as a complex network of energies in dynamic exchange. This perspective has been shaped through concurrence of different modes of thinking parallel to many scientific, technological and cultural progressions through centuries. Many different fields use computational methods with many distinct ways but specifically in the field of design, computation introduces a collection of multilayered concepts. These concepts are ranging from biology, mathematics, computer science, systems science, philosophy and etc., in addition to the coequal translations of such to the architectural design applications. The aim of this framework of concepts, perspectives and methods is to set a versatile yet comprehensive definition of what computational design thinking may constitute in architecture (Menges & Ahlquist, 2011).

To have a deeper understanding of computation and its compatibility to architectural design, the distinction between computation and computerization should be clear. The fundamental distinction can be broken down as methods in which *computation* extracts results from values or sets of values and *computerization* is simply just compiles given values. (Terzidis, 2006) One can increase the amount of information as well as the specificity while the other can only contain the initial information that has been supplied. This conflict is present since the computer has been integrated to architectural design. The essential issue concerning this thesis is the approach towards design rather than the particular skill sets or knowledge. In a computer-aided approach, containing information into symbolic representations which is the result of an object-based strategy is essential. On the other hand a computational approach contains values and actions in the form of codes in order to reduce the initial abstraction.

One crucial aspect of working in a computational manner is that it requires algorithmic information processing. *Traceable creativity* and search towards possible solutions are two essential features of the computational model -namely 'Adaptable Network Generator' (ANG) - proposed in this thesis therefore the topic of algorithmic design is studied and discussed with respect to the aims of this thesis.

There are many definitions of the word "algorithm" related to the field that it is used in. A general definition can be; *a set of procedures consisting of a finite numbers of rules, which define a succession of operations for the solution of a given problem* (Menges & Ahlquist, 2011). In this way an algorithm is a finite collection of instructions defined by the designer in an exact, complete yet general manner. The integration of algorithms into architectural design processes happens through computers, more specifically programming languages which enable computing. Working with computable and methodically developed architectural information sets is an alternative way to approach architectural design.

K. Terzidis summarize that notion as *Algorithmic Architecture.* According to Terzidis;

"*Algorithmic architecture involves the designation of software programs to generate space and form from the rule-based logic inherent in architectural programs, typologies, building code and language itself.*" (Terzidis, 2006).

Using scripting languages designers can extend their field of advantage and go beyond the factory set limitations of the present 3D software. Algorithmic thinking in architecture does not eliminate differences but combine computational complexity and intelligent computer use. Architects using algorithmic design techniques enable themselves to experience the shift from "*architectural programming" to "programming architecture".* This shift changed the priority of formalism and rationalism with intelligent form and traceable creativity. (Terzidis, 2006)

In theory if a problem can be defined under logical terms, a solution may be produced for the demands of the problem. An algorithm is a definition of a problem with different words. The linguistic articulation serves to communicate with the agent that processes the information. In the scope of this thesis this agent is to be a computer. An algorithm can be recognized as a moderator between the computer's ability to process and the human mind. One of the points this thesis underlines is that, this capability of an algorithm as a translator should be read as reciprocal. Both, as a way of describing to the computer how to proceed to solve the problem, and as a mirror image of the human thought in the form of an algorithm. (Terzidis, 2006)

Another way of approaching to the notion of algorithmic thinking is to see it as either a plan to solve a known problem strategically or a probabilistic search for the solution of a particularly known problem. In both cases it encodes the problem through rational, consistent and finite steps. Most algorithms are designed with a clear solution in mind to a specific problem but in some cases problems may have unknown or ill-defined solutions. In such cases algorithms becomes the way of exploring possible approaches that may lead to potential solutions. One of these approaches is called 'Generative Designing' which constructs the core of this thesis and will be studied in Chapter 2 in detail.

In addition to the 'Generative Designing' the second fundamental topic considering this thesis is 'Temporary Space Organizations'. Adaptive architectural design solutions are needed in case of complex spatial organizations such as temporary spaces. Consequently, parallel characteristics evolve in between the desired physical space, environmental stimuli and the shifting programmatic requirements. Physical spaces with such characteristics are formed through processes of *structural adaptations*. As a result they are no longer permanent entities (Busch, Ladurner, Baharlou, & Menges, 2011). In this thesis adaptable architectural design solutions are studied through temporary space organizations. Therefore fundamental features of temporary spaces and the computational methods for creating adaptable spaces in such contexts are researched in the realm of generative design, emergence and parametric design.

The computational model (ANG) developed in the scope of this thesis works in two fundamental levels – Analysis Layer and Generative Layer – therefore in addition to the above mentioned topics of 'Generative Designing' and 'Temporary Space Organizations' the topic of 'Spatial Analysis' is introduced.

1.2. Motivation, Objective and Scope of the Thesis

The motivation of this thesis comes from the desire to further investigate the question of “how generative systems can be applied to the design of temporal space organizations?" The question considered here has also been explored on a specific project in CompArch design studio (2010-2011) conducted in between TU Delft and METU. The main focus of this study is structured around a computational model called 'Adaptable Network Generator' (ANG) within the realm of computational design.

Temporal space organization is the main application area of ANG and one of the oldest topics in architecture. Through this thesis a computational model serving to develop ultra and infra-structure for temporal space organizations will be reinvestigated. The underlying advantages of the proposed computational model – adaptability, flexibility, responsiveness - are studied through several related discussions under the ‘temporary space organization’ topic.

ANG performs on the analytical data that comes from the multi layered analytical structure of its own. Therefore, spatial analysis methods and techniques are studied in order to structure the computational analytical tools used in ANG. As a result this thesis aims to describe the proposed computational model (ANG) in 6 chapters.

Chapter 2 covers the related literature of generative systems, parametric modeling, network models, temporary space organizations and spatial analysis methods.

Chapter 3 gives detailed information about the theoretical background of the ANG and methods chosen for the development of the program algorithm. In addition, it describes the two fundamental layers that form the ANG namely, spatial analysis layer and generative layer.

Chapter 4 focuses on goals regarding the usability of the model, the outputs of the model and tests conducted to validate and verify the goals regarding the usage of the model.

Chapter 5 introduces the case studies conducted in order to determine numerical and computational limitations of ANG.

In chapter 6, a brief summary is made and the thesis is concluded with the recommendations for the future work.

CHAPTER 2

A STUDY ON GENERATIVE SYSTEMS AND COMPUTATIONAL SPATIAL ANALYSIS WITH RESPECT TO TEMPORARY SPACE DESIGN

2.1. Introduction

The arguments of this thesis are developed according to the difficulties and insufficiencies of the methods and tools that has been used on author's graduation project in TU Delft in CompArch studio conducted between TU Deflt and METU. With respect to these insufficiencies, this thesis proposes a multi-layered computational model that contains different approaches from different disciplines that are relevant to the problem definition.

Design tasks determined by the project definition indicates that the proposed design solutions should be able to adapt different events and can be relocated for serving different locations in the given site. Such a project definition requires solutions in several different levels. In order to propose a working design solution, requirements of the project and topics related to this solution are clarified.

The design is expected to be assembled and disassembled for relocation which means that the aimed events to be hosted are temporary events. Therefore the topic of "Temporary Space Organizations" are studied. General characteristics and the analytical requirements of temporary spaces are investigated.

The proposed design solution is aimed to be a generative system that is capable of generating new design organizations for different locations and for different users. The design solution will be eventually a temporary space design but the results are expected to be the outcomes of a generative system. Therefore the topic of "Generative Systems" are researched.

The proposed design is aimed to be an adaptable component of the existing environment. In other words it is an agent with an intention to define relations between the components of a system and the system as a whole. With respect to this aim the topic of "Emergence" is studied as a subtopic of generative systems.

The adjustability of the outputs that are generated, according to the changing needs of the environment, is also expected as a fundamental feature of the proposed solution. For this reason the topic of "Parametric Modeling" is studied.

All the topics summarized above with their reasons to be studied are the concepts that constructs the theoretical background of the proposed computational model in this thesis. These concepts and their reflections on the proposed model requires a multi-layered framework to function properly. All these layers are expected to be functional individually, as groups and as a whole. In order to understand the required relations among those parts "Network Models" and their applications in different fields are investigated.

The final component of the proposed multi-layered computational model is the analytical layer. In order to understand the qualities of the required analytical tools with respect to the aims of this thesis and the above mentioned graduation project the existing computational spatial analysis tools and methods are researched and studied. With respect to the knowledge gained from the conducted research on "Spatial Analysis Methods" the required analytical tools are decided for the analytical layer of the proposed computational model.

As a conclusion this chapter studies the topics related to the proposed computational model with respect to the difficulties and insufficiencies of the methods and tools that has been used on author's graduation project. In order to clarify and position the argument inside the architectural domain the topics of temporary space organizations, generative systems, network models and spatial analysis methods are studied.

2.2. Temporary Space Organizations;

Temporary space design is one of the oldest topics in the realm of architecture. In this study the fundamental features of temporary space design and the relations of such spatial organizations with their environment are discussed. In the scope of this thesis temporary space organizations are the main focus for the proposed computational model. Therefore temporary space design is studied over the concepts of mobility, stability, event-facility space and environments with changing needs.

Temporary organizations are event-facility spaces. They occupy a part of the urban landscape for a short and specific amount of time supplying quality space for the current event. Urbanity and architecture is a relevant ground for the permanent/transient polemic with respect to temporary space organizations. They can be conceived by developing a conceptual system in which the physical organization of event-facility spaces are separated. Event spaces are programmatic platforms without any formal boundaries. They are formed as '*free radicals*' that can flow through their surrounding creating new configurations and possibilities. (Kronenburg, Joseph, & Wong Y, 1989) This conceptual approach of space making is inspired by environmental behavior, than form making. This shift in the design approach to urban landscape has a potential to generate new event-facility patterns with new morphologies for different urban locations. The emergence of these event-facility spaces does not necessarily contradict with the existing historical urban fabric. In fact, the superimposition of these two flows is crucial as they can be reintroduced to the new facility matrix of the city.

The notion of motion is a fundamental part of architecture. Everything that is in motion is relative to something that is static or in motion. Kinetic motion is now an essential part of the architectural thought and realization. This idea of motion articulates itself to architecture as a process of growth. As a fact, architecture is no longer static on the contrary it should be understood as extendable, adjustable and expandable in time as well as space. (Müller-Schloer, 2004)

This does not mean that stability is no longer a relevant notion for urban and architectural environment. However, there is a difference between 'static' and 'stable'. Stability has a particular role in both architectural and urban terms as a supporting platform for motion. The artificial landscapes we created need defensive elements against entropy which serve as strategic platforms for mobility. "*Every mobile artifice springs in some way from a fixed material base, cars rely on roads, radio waves on a mast, planes on a runway*" (Kronenburg, Joseph, & Wong Y, 1989). There is a direct relationship between infrastructure and mobility. Therefore temporary space organizations generally need a static platform in order to be functional.

The decision making process regarding the features of a temporary space design requires an understanding on environmental patterns and their characteristics. These patterns can be understood as temporary phenomenon which are continuously responsive to the environmental flows, fluxes and rhythms. This interactive reformation creates a multitude of space, time and objects. In such an interactive and responsive habitat these strategic patterns are formed only relative to a particular moment in time. One can interpret these patterns as a language of environmental flows and construct theories in relation to the flows of information from the environment. One can overcome the complexity of such systems by abstracting and formulating theories from environmental pattern recognition and increase the control successfully.

These theories can be considered as the results of an emergent process which emerges from topography in response to the programmatic forces of a holistic matrix. This strategic, formal redistribution is a reflection of the local environment where programmatic requirements are laid down and adjusted over time. As a result these patterns can be seen as sets of interrelation matrices in other words three dimensional patterns within a flow of programmatic forces that are influencing both the emerging and emerged forms/organizations constantly. Temporary existence of things is the core of a city form. Gary Brown describes it as; "*a single frame in the animated growth of appropriate solutions from one super-positioned state to another.*" (Brown, 2004) This movement reveals patterns in a city that makes its unique existence so unique. Form is temporal and transient that's why the space and the way it emerges is a more desirable design reference for this thesis.

In this context this thesis the proposed computational model supplies possible solutions for infrastructure (under surface) in addition to the adjustable organizational solutions for the space design (on surface). It aims to generate network designs that create interactions between different components of the given context. These components are dynamically responsive to the urban environment. This means that any change in the environment triggers a bidirectional flow resulting in a new space configuration (on surface) which directly affects the infrastructure (under surface). Without a working infrastructure, temporary organizations fail to exist.

A computational approach towards the concept of infrastructure organizations is under exploration for some time. Different algorithms are being used to generate optimized infrastructure models to generate traffic flow of a city, subway networks, cell phone transmission tower distributions, etc. The level of the computational tool for generating infrastructure paths that is proposed in this thesis is not as complex as a subway network. It is a proper implementation of the *shortest path* algorithm in order to optimize the infrastructure for a multi-cellular temporary space organization.

In the scope of this thesis space creation is studied from a computational stand point. Considering that there are many distinct ways of space creation, ANG isolates itself from the conception of form and focuses on the circulation and organization of the spaces. Therefore with respect to the fundamental features of the above studied topics of temporary space design, the methods of spatial analysis are studied in order to supply the required information for space creation in environments with changing needs.

2.3. Generative Design

Generative design, is one of the substantial topics of this thesis. The discussions conducted in relation to the topic starts with the early studied of Alexander and continues with an inquiry on contemporary approaches.

The notion of *unselfconscious process was* found in the early writings of Christopher Alexander. Fundamentally this refers to a reciprocal relationship in a social context where environment, building and culture are formed simultaneously. Christopher Alexander extended this notion into;

"How architectural problems may be solved through an analogous process where design forms through the iterative readings and responses to interrelation conditions, with the intention of producing environments synchronous with their cultural settings." (Alexander, 2011)

The significant point that Alexander emphasized was that the conception of form is no longer observed as an object instead it is an externalized operating system.

In the scope of this thesis generative systems are studied in order to answer the question "how generative systems can be applied to the design of temporal space organizations?" In his article published in 1968 (Architectural Design) Alexander started the base argument of a generative system which is very parallel to the argument of this thesis. His question was *-how such a system is born, itself, of a generative system-* setting up a duality between the object as a computing agent and the method as a computational process. Alexander points out some important prospects of such a process, describing three integrated conditions; *-the global behavior, the components that form such a behavior and the types of local relationships among those components.-* These three aspects lay out the characteristics of a system but it is not described how such a system can be generated. Alexander clarifies this by explaining four fundamental features of a system; the difference between a group of actions forming the behavior and the system that generates the behavior, as a series of interactions, as well as the level of abstraction needed to understand such a complexity.

1. There are two ideas hidden in the word system: the idea of a system as a whole and the idea of a generating system.
2. A system as a whole is not an object but a way of looking at an object. It focuses on some holistic property which can only be understood as a product of interaction among parts.
3. A generative system is not a view of a single thing. It is a kit of parts, with rules about the way these parts may be combined.

4. Almost every 'system as a whole' is generated by a 'generating system'. If we wish to make things which function as 'wholes' we shall have to invent generating systems to create them. (Alexander, 2011)

Although the explanation of Alexander's on the fundamentals of a generative system is one of the earliest studies on this topic, it is still a valid reference in the scoop of this thesis. In addition, his motivation on generative systems creates a solid ground for the 'adaptable network generator' research which substantially is a generative system capable of interaction and adaptation to the environmental changes and components.

2.3.1. What is generative design?

Generative designing differs from conventional design processes in many ways. It is the use of a combination of several mathematical methods in order to generate possible, alternative set of solutions for the current design problem. The use of generative techniques gives us possibilities to solve complex problems for which conventional approaches may not give any plausible solutions.

The contemporary developments in technology and computer science have influenced many fields including art and architecture. In art generative design methods are used in order to create complex works of art. Galanter's definition on generative art can be considered as a valid definition to understand the nature of the process.

"Generative art refers to any practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art." (Galanter, 2003)

Galenter's definition can also be applicable for architecture. In many cases a computer program (script) is used. These programs can have predefined rules or can contain new rules defined by the designer for the problem at hand. Never the less these programs have similar underlying design patterns:

1) Describe the initial state;
2) This state will be modified according to a well describe set of rules;
3) Evaluation of constraints; these constraints test if alternatives fulfill the design goals;
4) This new state, derived in step 2, will become the initial state for the next iteration.

This iterative process can be fully automated, in which case the designer determines when to terminate the process. However, a single or multiple steps can be completed by the designer himself, in which case the designer becomes part of the iteration process.

The procedure mentioned above contains two basic parts namely; "*state*" and "*rule-set*". (van der Zee & de Vries, 2008) The state represents architectural design; the rule set expresses the transformations and describes the constraints. The design should fulfill these constraints in order to be an acceptable solution. The initial state will be transformed into alternatives of the primary design, depending on the degree of freedom. This formal approach shows a big similarity with theoretical and practical design methods as well as practical methods. These complex require high computational power which may not be provided by contemporary design approaches, due to computational time and capacity of human mind. Generative systems help designers to benefit the number crunching power and calculation speed of the computers to solve such design problems (van der Zee & de Vries, 2008).

2.3.2. Emergence as a multi-layered role model;

Emergence is a very important subject for architecture, demanding significant revisions to the approaches in design processes. In addition it is a crucial part of the generative systems and strongly related to the aims of the computational model proposed in this thesis. This computational model is designed as a digital platform for designers to analyze and process the information flows of the selected environment. This process basically works by defining the relations between the components of a system and the system as a whole. This relational framework can also be found in emergent systems as; the interaction between macro-level and micro-level. In order to have better understanding on the shared features of the proposed computational model and emergent systems, the concept of emergence is studied.

Environments with changing needs and conditions have collective behaviors. This collective behavior emerges from the periphery of interactions within a set of singular components. These interactions can happen in many different scales from molecules and cells to ecosystems and climates. The complex is heterogeneous, composed of many varied and interdependent parts, which all behave uniquely based on encoded processes and specific environmental conditions (Hensel M. M., 2010).

Complex systems found in nature usually contains two generative systems namely emergence and self-organization. De Wolf and Holvoet defines emergence as:

"A system exhibits emergence when there are coherent elements at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such

emergents are novel with regards to the individual parts of the system." (De Wolf & Holvet, 2006)

On the other hand the definition De Wolf and Holvet proposes for self-organization is;

"Self-organization is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control." (De Wolf & Holvet, 2006)

Emergence and self-organization are both dynamic processes that arise overtime. They are powerful processes which mean that they have the ability to overcome occasional failures of single elements. The only possible way of obtaining a consistent behavior at the macro level is to let that behavior arise and organize autonomously. If the aim is to reach a level of consistency in a complex system, combination of these two phenomena is a promising approach.

Moreover, an emergent process should be adaptive in order to host a system that is capable of self-organization in a state of environmental flux, where the system responds to changing stimuli. Local conditions influence the entire systems and each element arises in relation to these conditions. (Herr, 2002) Adaptation can be seen as a result of parallelism and iteration in a competitive environment with finite resources. (Flake, 2000) In this instance behavior is the case subject of fitness which determines a reaction for or against particular actions based on feedback mechanisms at all levels. It is specifically useful to consider adaptation in complex systems, because it simplifies the control on complexity by letting the complex to evolve based on rules that can relate to practical applications. (Flake, 2000)

The composition of an emergent system with a multi-layered structure, have been taken as an example while designing the structure of the proposed computational model. Therefore, working mechanism of emergent systems and the notion of coherence in in such systems are studied.

The definition of emergence uses the concept of 'emergent' as a general term to indicate the result of the process: properties, behavior, structure, patterns, etc. The 'level' addresses to specific points of view. The macro-level assumes the system as a whole and the micro-level assumes the system from a perspective where the individual entities generate the system. In other words, the global behavior of the system (i.e. the emergent) is the consequence of interactions among the individual components of the system. The collective behavior is, however, contained in the behavior of the parts under the condition of being studied in the context they are found. Properties of an emergent process cannot be studied by removing parts from the whole and examining them individually. However, they can be studied by observing each part in the context of the system as a whole.

Coherence is an important part of an emergent process referring to a logic and consistent correlation of parts. Emergents tend to create a consistent sense of identity over time as integrated wholes (i.e. persistent pattern). Coherence covers and correlates the individual lower level components into a higher level unity, a coherent whole. This coherence is also called '*organizational closure'*. (Heyligen, 2012) In systems with emergence, emergents arise in time as the system evolves. Therefore, the appearance of the emergents should be considered as a dynamical construct and are related to the appearance of the new attractors in dynamical systems. In emergent systems the macro-level and micro-level are connected with a bidirectional link. The parts in the micro-level generate an emergent structure on the macro-level. In the opposite direction the emergent structure has an influence on its parts as well. (De Wolf & Holvoet, 2004) The features studied above on emergent systems are internalized and interpreted in the realm of this thesis in order to provide a multi-layered structure for the proposed computational model.

Form and behavior emerge in relation with the process of complex systems which are continuously and dynamically in interaction with the environment. Forms in relation to the environmental patterns maintain their continuity and integrity by changing features of their behavior and by iteration over many generations. Forms exist in many different populations, and where communication between forms and the environmental patterns effective, collective structured behavior and intelligence emerges. (Hensel, Menges, & Weinstock, 2004) The emergent systems which generate forms and the systems within complex forms themselves are provided by the flow of energy and information through the system. These flow patterns have constant variations, adjusted to maintain the balance by 'feedback' from the environment. An emergent whole form can be a component of a system which is a sub-system of a higher level one. In other words a 'system' for one process can be 'environment' for another.

"We are within the horizon of a systemic change, from the design and production of individual 'signature' buildings to an ecology in which evolutionary designs have sufficient intelligence to adapt and to communicate, and from which intelligent cities will emerge". (Hensel, Menges, & Weinstock, 2004)

The task for architecture is to describe a working understanding of emergence and to outline the mathematics and process that can make it useful to us as designers. This means that we must study and search for the principles and dynamics of organizations and interactions in order to set mathematical rules that can be utilized by artificially constructed systems as the one proposed in this thesis.

Emergence in this perspective is a very important subject for architecture and for this thesis. It offers a new perspective to the existing design approaches. Designers can use computational models like ANG that can create connections with the environmental flow and fluxes within computational environments to perform spatial analysis and generate multi-cellular organizations. Intelligent environmental behavior of temporary spaces can be much more effectively produced and maintained by the collective behavior of the organic, emergent systems. This notion must be extended beyond a single individual building/space design and its response to environment where each building/space is part of the environment of its own and its neighbors.

2.3.3. Parametric Modeling;

Parametric modeling is known to be a tool to produce adjustable digital models to explore design possibilities but it is also a way of thinking and a practical approach for *traceable creativity*. In this thesis the case studies discussed are produced by parametric modeling tools. In addition, the proposed computational model contains two new algorithms developed as a component for the parametric modeling software Grasshopper. Therefore fundamentals and advantages of parametric modeling are studied.

Parametric modeling CAD systems are becoming standard tools in the architectural domain to aid the design process in research, academy and practice. Parametric modeling systems are initially developed with an intention to aid aeronautical industry but currently they are making their way into the architectural domain as well. This alternative approach is being accepted since it provides a strong and applicable framework for the conception of the design, enabling the definition of multiple instances and possible design solutions from a single modeling schema. (Barrios, 2005) Parametric modeling systems are proposing a challenging contribution or even a shift to the traditional way of computer use. Traditional use of computer systems is aimed for representative purposes, on the other hand parametric systems offer intelligent models capable of interacting and responding with the local and global variables affected by the environment and by the designer at will. Carlos Barrios explains the expansion of the current limitations of the traditional CAD systems by the effect of parametric modeling systems as;

- *Offering more flexibility to design parts and assemblies of complex nature;*

- *Provide reliable systems to test instances of designs from a single model;*

- *Expand design exploration of at the initial stages of the process.*

Parametric modeling requires a precise thought and a clear mind in the process to build a model which can fulfill the requirements of the designer even if it has a very sophisticated structure. This task can be thought as a time consuming processes but a good parametric model has the advantage to provide a solid structure that will act as a container of information of the design history. Furthermore, the model has the potential to be flexible enough to be constantly evaluated, revised and updated if different components are changed, added and deleted within the same structure of the parametric model.

In many parametric models like the model proposed in this thesis, some elements of the system have attributes that are fixed and some attributes that can vary. The attributes that are fixes are called explicit and the attributes that are subject to change are called variables. Through a process of parameterization the explicit attributes become variables. This process defines the variation patterns of the model in other words it defines which components will vary and how the variation occurs. Variables can be independent with a value or a set of values attended to them. On the other hand variables can be constraints to a particular range of values which means that they can be dependent in which the value of the material is linked or related to the value of another entity of the model. (Barrios, 2005)

2.4. Network Models

Network models have many applications regarding many different fields including architectural and urban design. A generative model employed in the definition of *Network Design*, is the main focus of this study. Here in this chapter the features of the proposed network will be studied as well as the definition of "network" and the idea of network in different fields with respect to the aims of this thesis.

The basic definition of network without referencing any field is; a large system consisting of many similar parts that are connected together to allow movement or communication between or along the parts or between the parts and a control center. (Thesaurus, 2014) This definition differs depending on the field although the main notion remains the same.

The network idea in biology suggests that any system with sub-units that are linked into a whole is a network. Complex biological systems may be represented and analyzed as computable networks. For example, ecosystems can be modeled as networks of interacting species or individuals. Similarly, a protein can be modeled as a network of amino acids with nodes and edges. Amino acids can be represented as a network of atoms such as carbons, nitrogen and oxygen. (Proulx, Promislow, & Phillips., 2005)

A parallel definition to biological network is neural networks which uses much more computational models through its process. This topic is also studied under ‘spatial analysis’ section from a different perspective. The term neural network was traditionally used to refer to a network or circuit of biological neurons. (Hopfield, 1982) The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes.

A computer network, or simply a network, is a collection of computers and other hardware interconnected by communication channels that allow sharing of resources and information. Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network. (Akyildiz & Rudin, 2014)

Urban networks (transportation networks) are very similar to the proposed model in terms of application of the computational technique. On contrary the rest of the network models that has been discussed have theoretical similarities. Many urban networks use graph theory to simulate and visualize traffic flows. In graph theory, a flow network (also known as a transportation network) is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. Often in Operations Research, a directed graph is called a network, the vertices are called nodes and the edges are called arcs. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, except when it is a source, which has more outgoing flow, or sink, which has more incoming flow. A network can be used to model traffic in a road system, fluids in pipes, currents in an electrical circuit, or anything similar in which something travels through a network of nodes.

The most related type of network regarding to the scope of this thesis is spatial networks. A spatial network is a network of spatial elements. Spatial networks are derived from maps of open spaces within the urban context or building. Space maps are known as the negative images of the standard maps with the open space cut out of the background buildings or walls. (Hillier & Hanson, 1984) The space map contains smaller units like road segments. A common use of a spatial network is transportation network analysis. In a graph representation the nodes are street intersections and the edges are the road segments.

For practical applications, the space symbolizes the two-dimensional space and the metric is the standard Euclidean distance. Transportation and mobility networks, Internet, mobile phone networks, power grids, social and contact networks, neural networks, are all examples where space is relevant and where topology alone does not contain all the information. Characterizing and understanding the structure and evolution of spatial networks is important for many different fields as well as the computational model developed in this thesis. (Barthélemy, 2011)

The proposed computational model in this thesis also processes information. In that model the data is derived from local environment. The properties of the design problem may affect the level of interaction with the surrounding but the network design is aimed to connect these variables. Another similarity between the networks generated by the proposed model and the examples explained above is its nature to simplify and structure the complexity. Parallel to the working mechanism of the neural networks, the proposed model also abstracts the data from its environment. In our case the purpose is to transfer the information into mathematical or geometrical data with an intention of creating computationally processable data.

The fields mention above are just some examples where the concept of network is used. The important point is that all these networks are used to help to structure the complexity of their fields and to analyze the relation between parts of a whole. The ANG on the other hand helps designers to generate possible design solutions in different contexts. It is not only an analysis tool to understand a specific case instead it is a generative model that uses the data from analysis to create alternative solutions for the changing environmental demands. The nature of these changing environmental conditions is directly related to the dynamic urban patterns in a city. In the scoop of this thesis the dynamic urban patterns are studied adopting an organic system approach. Therefore organic systems theory is studied in comparison to mechanical systems theory under the temporary space organizations topic.

2.5. Spatial Analysis Methods

Developing technologies and computer power are strongly influencing the way we design. Computational methods and tools are being used in the architectural profession and researh more than ever. In order to structure a working computational approach one needs to supply the required information for computer to process. Some part of that information comes from the analysis phase of the design processes. Here in this chapter the computational space analysis methods used in architectural and urban design namely; Space Syntax, Cellular Automata and Graph-base Analysis are to be studied. In the conclusion a multi layered spatial analysis method will be discussed with respect to the aim of this thesis.

2.5.1. Space Syntax

Space syntax is a term that is used to describe a family of theories and techniques concerning the relationship between space and society. It emerged from a dynamic and active research group based at University College London, in the early 1970s, and led by Professor Bill Hillier. The original driving force behind space syntax research was prompted by a goal to understand the relationship between space and society (rather than space and an individual subject) (Hillier & Hanson, 1984). Initially it was thought that by holding a 'spatial configuration' to be an artifact of the society that constructed it, then by studying such a system of spaces, it should be possible to more fully understand the society itself (as would be true of studying any other kind of artifact). However, the relationship between space and society is a two-way relationship: not only does a society create the spatial systems that it uses, but a group of people (be it the inhabitants of a settlement, an urban neighborhood or the users of a complex building) is directly affected and influenced by the spaces they inhabit. In particular, one extremely powerful way in which a pattern of spaces, or a configuration, affects its users is through pedestrian movement. Any set of spaces, of sufficient complexity to be described as a configuration, forms a spatial hierarchy in which some spaces become more strategic and others less so. These strategic or, on average, more accessible spaces will tend to attract a higher rate of pedestrian movement than other, more segregated, spaces. This is clearly explained in Bafna's excellent introduction to space syntax (Bafna, 2003). One of the key methods of analysis used in space syntax research is a graph-based technique that is able to identify and represent this varying pattern of more-to-less strategic spaces. Although such techniques were not originally developed as tools for predicting pedestrian movement, it has been found that there does exist a powerful relationship between movement and spatial structure and so this graph-based analysis may be used to predict relative rates of pedestrian flow. It is this predictive ability of space syntax analyses that has caused it to be adopted as a design tool by many architects and urban planners (Dalton, Hölscher, & Turner, 2012).

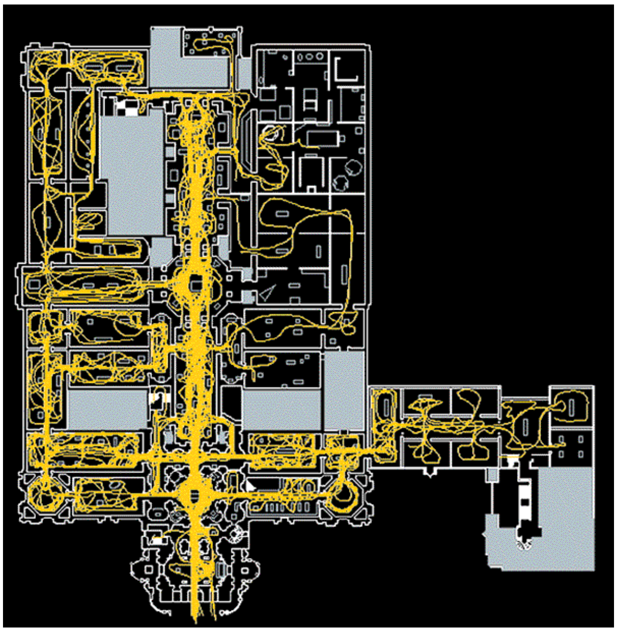


Figure 1: Space syntax application for predicting active walker patterns, illustrated by Helbing, Keltsch and Molnár (/http://www.nature.com/nature/journal/v388/n6637/fig_tab/388019a0_F1.html Last visited: 28.1.2014)

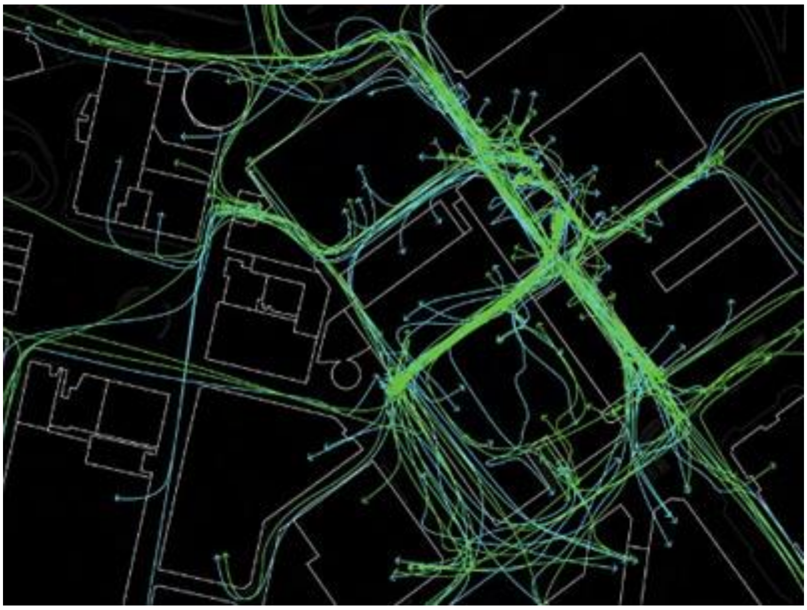


Figure 2: These patterns include movement on foot, on cycles and in vehicles; way finding and purchasing in retail environments (http://www.visualcomplexity.com/vc/project.cfm?id=692 Last visited: 28.1.2014)

2.5.2. Graph-based Analysis;

In the field of architectural and urban design, graph based techniques are commonly used as representations of space to study and analyze spatial behavior. In mathematics and computer science a graph is an abstract construct consisting of objects called nodes and their relations implemented in edges. Many real-world problems of practical interest can be efficiently represented using graphs as a flexible, extendable, and generic framework offering various straightforward algorithms for the solution of specific higher level questions. Both in architecture and cognitive science mathematical graphs are used to systematize the description of the human spatial environment. (Hübner & Mallot, 2002)

In spatial cognition and artificial intelligence, graphs have been used as models for mental representations of environments for decades. First and most importantly, graphs are models of the mental representation of environments, serving as working hypotheses for the structure, format, and content of spatial memory which can be empirically tested. Furthermore, graphs can describe the set of movement actions available at a given place.

The particular appeal of graph structures as models for spatial memory arises from their increased flexibility as compared to map-like representations of space. For example, while basically being topological structures, by labeling or weighting single edges of graphs, distance and direction information is included that allow for metric navigation abilities such as short-cutting behavior (Hübner & Mallot, 2002). Additionally, various non-spatial information can be attached to the nodes, for example, places can be labeled with emotional or episodic information (Fellous & Arbib, 2005).

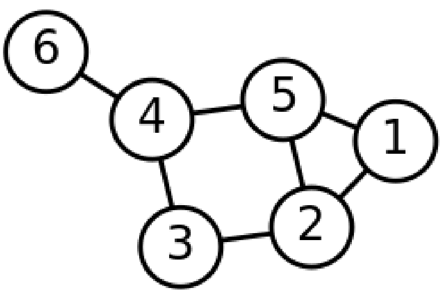


Figure 3:Drawing of a graph

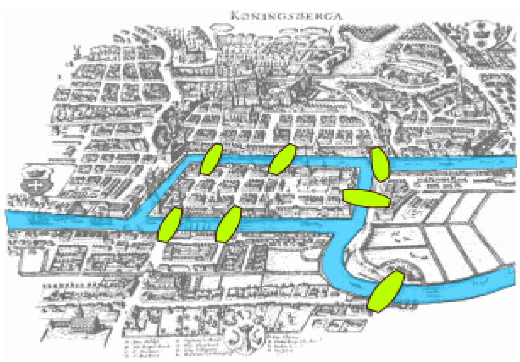


Figure 4: The Königsberg Bridge problem (Hübner & Mallot, 2002)

The first paper on graph theory was written in 1736. It was a study on the Seven Bridges of Köingsberg by Leonhard Euler (Figure). Since then many scientist worked on graph theory in many different fields. *Graphs c*an be used to model many types of relations and process dynamics in physical, biological and social systems. Graph theory in computer science and topological graph theory are the most related fields to be studied with respect to the context of this thesis. Basically, in computer science graphs are used to represent networks of communication, data-organization computational devices etc. On the other hand typological graph theory studies the embedding of graphs in surfaces, spatial embedding of graphs, and graphs as topological spaces. (Gross & Tucker, 1987) The main reason to study graph based analysis comes from the generative features of the proposed computational model. In the context of this thesis being generative means that the designed system can adapt different contexts and environments with changing needs. Consequently such an approach requires a great deal of analysis in different levels.

That is why graph base analysis are to be studied in comparison to each other in order to construct a general understanding and ground for the hybrid spatial analysis method for temporary space organizations that has been proposed in this thesis.

2.5.3. Graphs in architectural analysis

Graph applications in architecture have a long tradition based on graphical or diagrammatic analysis and have been substantially influenced by the phenomenal city descriptions of Lynch (1960). The need for strictly formalized description systems arose from the wish to do quantitative comparisons between spatial configurations in order to identify the essential properties in terms of function or usage. In the domain of space syntax analysis, spatial organization patterns were seen as close parallels to the underlying social structures (Hillier & Hanson, 1984). Besides applied research, graph investigations in architecture particularly concentrated on methodological issues such as the transfer of analysis techniques on arbitrarily shaped environments or on variable scale levels and on the formalization and automation of the graph generation process. Also approaches to determine and minimize the number of necessary nodes were explored (Bafna, 2003).

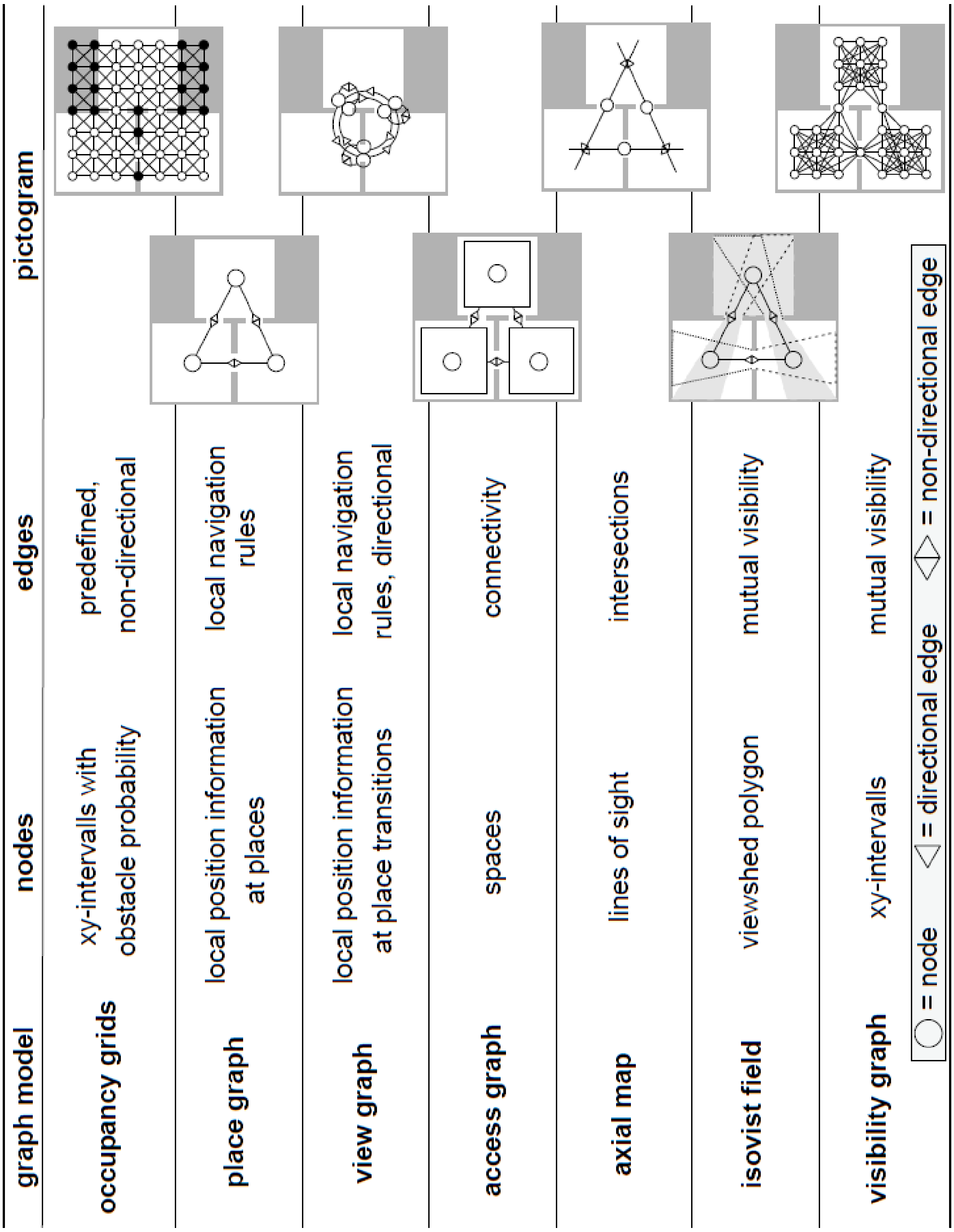


Figure 3: Overview on the different graph models.
(Hübner & Mallot, 2002)

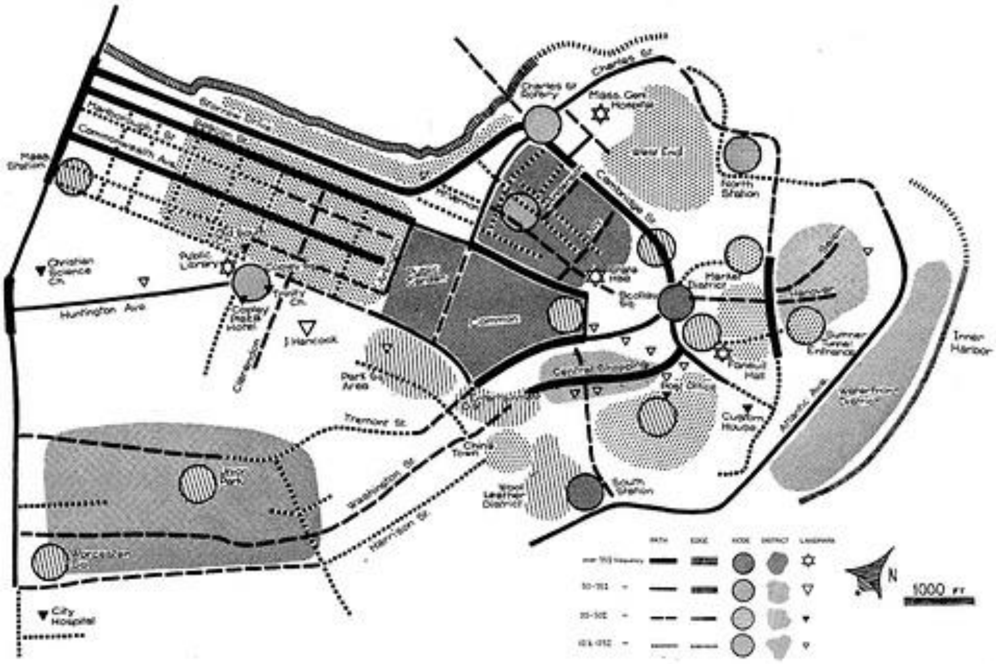


Figure 4: Kevin Lynch, Image of the City (1960)

As visualized in Figure 4 Lynch studied the conventional maps with graph based techniques to create *mental maps* of a city with five elements. In his work *paths* represent; the streets, sidewalks, trails, and other channels in which people travel, *edges* represent; perceived boundaries such as walls, buildings, and shorelines, *districts* represent; relatively large sections of the city distinguished by some identity or character, *nodes* represent; focal points, intersections or loci, *landmarks* represent; readily identifiable objects which serve as external reference points. (Lynch, 1960) Since the work of Kevin Lynch, graph based analysis have been used by many architects and urban planners for many different purposes. These methods and techniques which can be seen in figure 6 namely; occupancy grid, place graph, place graph, access graph, axial mapping, isovist field, visibility graph are to be studied.

Occupancy grid; in artificial intelligence, occupancy grids are often used as representations of space for autonomous navigating robots. In occupancy grids the environment is mapped on a regular array of cells, which can be conceived as a specific graph implementation. Each cell is connected to its eight surrounding neighbors and holds a probability value that the cell is occupied by an obstacle (Moravec & Elfes, 1985).

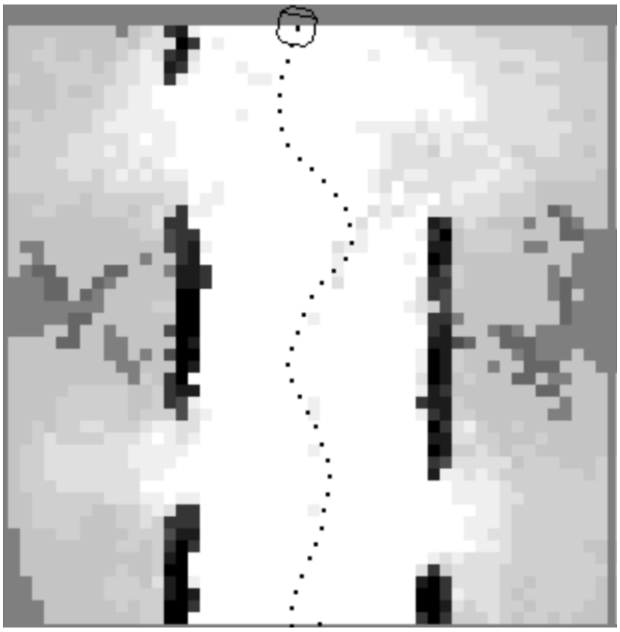


Figure 5: An early study of Occupancy Grid Mapping (1997).Results of the MURIEL algorithm on a typical hall scene. Grayscale indicates occupancy, with white being unoccupied and black being occupied. For calibration, the darker gray areas on the center far left (Hübner & Mallot, 2002)

Occupancy grids do not represent selected places but are continuous representations. The advantages of these structures are their simplicity and their metric embeddedness. However, occupancy grids often suffer from their directional inhomogeneity and their general structural rigidity.

This method is used to generate simple regular grid-based maps (generally) for robots to navigate through a space with obstacles. The fundamental features match with the 'adaptable network generator' in terms of user navigation but the subject to that method is a semi intelligent robot. In the case of ANG the users are human beings with high intellectual capacity with genetically coded navigation skills. This method focused on avoiding obstacles; on the other hand ANG deactivates grid cells with obstacles. In addition, the occupancy grid mapping does not support working with irregular grid cells. So occupancy grid mapping is a promising method for its field but does not meet the needs of a temporary space organization and dynamicity of an urban environment in terms of navigation.

Place graph; in the place graph concept, nodes correspond to single places or positions within an environment, edges describe the connectivity between nodes. In their most basic form place graphs are purely topological representations of space, in which nodes carry local position information, allowing the identification of the corresponding place. Edges carry local navigation rules, such as 'turn left' or 'follow road', that allow navigating between nodes (Hübner & Mallot, 2002). Kuipers (2000) and Kuipers, Tecuci, & Stankiewicz (2003) have suggested a slightly different concept. In this bi-partite graph both places and paths between places are represented as nodes that are linked together by edges. While both of these concepts are in their basic form topological representations of space, metrical information such as distance and direction might be associated. In contrast to occupancy grids, place graphs represent selected places or positions within the environment rather than the environment as a whole and can therefore be conceived as more sparse. (Kuipers, Tecuci, & Stankiewicz, 2003)

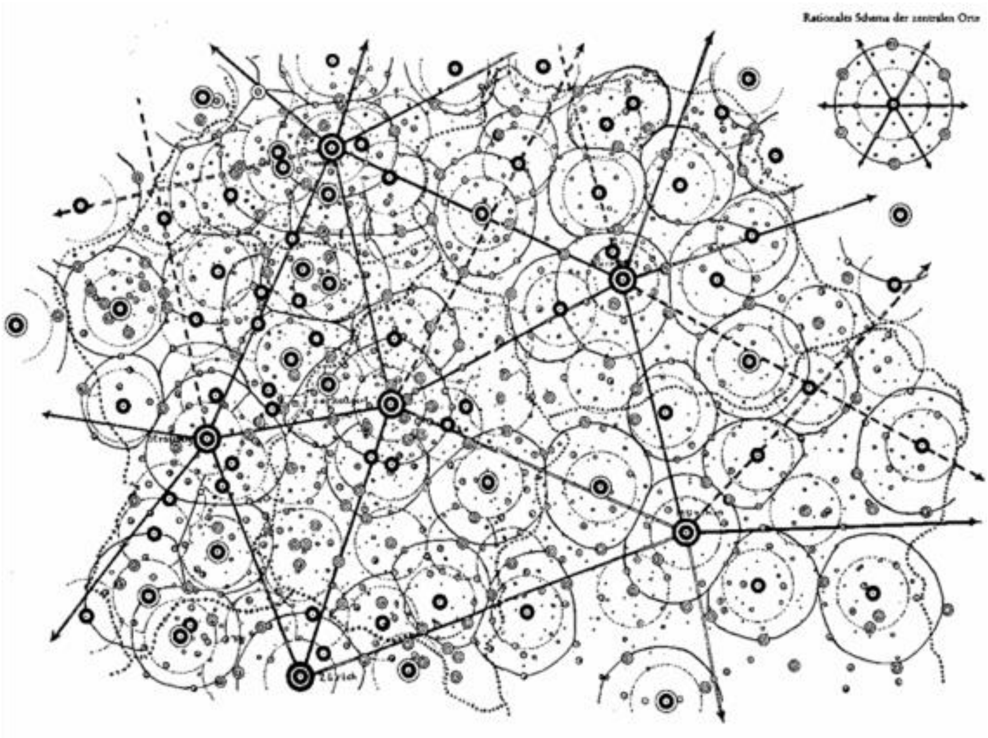


Figure 6: A place graph application with respect to the Central Place Theory. Christaller's model produces a hierarchy of increasingly spaced centers at different scales. Walter Christaller (1933)

Place graph concept is quite parallel with the intentions of the 'adaptable network'. The nodes in a place graph identify the location of a cell in its environment (in the local urban grid). This simple rule of placing nodes to the center of each or related cells create a plain but strong ground for multi-level, more complex emergence. In the concept of place graphs the edges connecting nodes are for navigation purposes though in 'adaptable network' edges may contain more information than navigation like infrastructure paths or incremental growth directions.

View graph; Schölkopf & Mallot (1995) have proposed a minimal spatial memory model in which each node corresponds to a pictorial snapshot of the environment as seen when navigating a given place transition. Nodes are connected by edges if the corresponding views can occur in immediate sequence while walking through the environment. They are labeled with local navigation rules. The basic idea of the view graph is to generalize route memories given as chains of recognition-triggered action sequences to a more flexible yet still parsimonious representation of space allowing for complex navigation behavior such as route planning. View graphs have been successfully used for robot navigation (Mallot, Franz, Schölkopf, & Bülthoff, 1997) and for the explanation of human navigation behavior (Gillner & Mallot, 1998; Steck & Mallot, 2000).

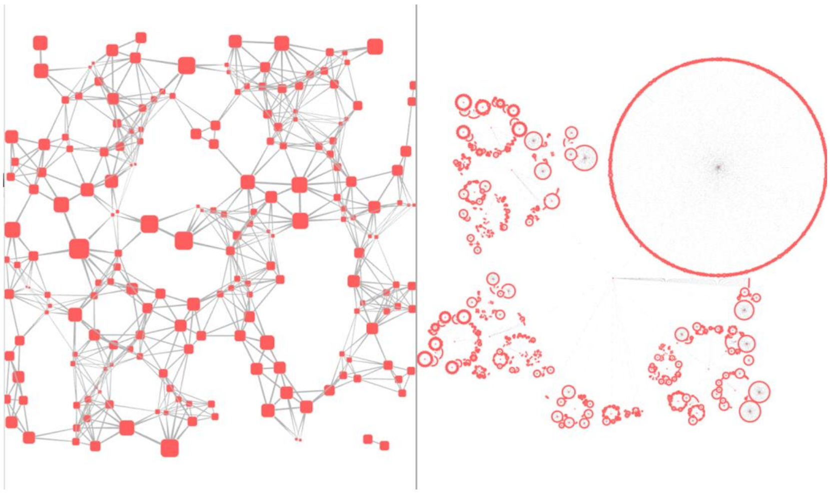


Figure 7: A screen shot of a View Graph from the software "Tulip". (http://tulip.labri.fr/TulipDrupal/?q=news Last visited: 28.1.2014)

View graphs share more or less the same goals and application fields as occupancy graph maps. They are both used in field of robotics for navigation but view graphs can handle more complex data. This means that in a way this method supports more intelligence. View graphs work with pictorial snapshots which are represented by nodes. So in terms of node based representation they highly differ from the approach of 'adaptable network'.

Access graph; One line of research focused on inter-cultural comparisons as well as on the application in architectural practice and therefore pursued the elaboration of improved generic graph descriptor variables (e.g., capturing the connectivity, centrality, control level of places). (Hübner & Mallot, 2002) Early space syntax analyses (Hillier & Hanson, 1984) made use of phenomenal spatial units such as clearly defined rooms or labeled places for their nodes, while graph edges binarily signified their mere connectivity. As additional graph element, the accesses to the individually analyzed spatial configurations were considered as root nodes of the so-called justified graphs.

Access graphs on their own are simple analysis methods for visualizing the connectivity of spaces. The nodes represent spaces and edges represent connectivity. This method cannot be used as a standalone analysis method for the ‘adaptable network’ but it can be used as one of the levels of the final multi-level hybrid analysis method.

Axial maps; consist of nodes describing lines of sight or straight movement and their mere intersections as binary edges. They are based on a prior partitioning of the underlying environments into a near minimal set of convex subspaces. In a second step, these convex hedras are connected by the smallest possible number of straight lines of maximum length.

The generated graph has to meet the requirement that each adjacency of the subspaces can be associated to at least one axial line. A recent extension of axial maps is angular maps that additionally consider the angle between the axes in the connectivity edges. Axial maps have been mainly used for the analysis of city quarters. Strong correlations between derived descriptor variables and statistical pedestrian dispersal have been found.

Axial mapping method works almost the same as access graphs. The only difference is axial maps are based on lines of sight. The concept of line of sight is an important issue for the ‘adaptable network’ because the visual connectedness with the surrounding context (existing buildings, historical buildings, monuments) is one of the parameters that shape the computational model in urban environments. Therefore axial mapping method can be used for analyzing visual connections of the grid cells of the ‘adaptable network’ and the surrounding buildings or objects. (Franz, Mallot, & Wiener, 2005)

Isovist field; For analyzing spatial characteristics of smaller environments, Benedikt (1979) has proposed isovists as objectively determinable basic elements. Isovists are polygons that capture spatial properties by describing the visible area from a given observation point and therefore lend themselves particularly well for analyzing open-plan indoor spaces. In order to describe spatial characteristics of environments beyond a single sensory horizon, isovists can be used as content in graph nodes and connected by intervisibility edges (Franz, Mallot, & Wiener, 2005).

Isovist analysis are also been used in urban scale in order to study the visual connectivity of singular or multiple destination point from a central observation point. Space syntax analysis and isovist analysis have been integrated, more recently, in *visibility graph analysis*, which connects together mutually visible points in a graph data structure (Turner, Dox, O'Sullivan, and Penn, 2001).

Visibility graph; Derived from isovist fields, Turner, Doxa, O'Sullivan, & Penn (2001) have proposed visibility graphs as a promising way to optimize the computational graph analysis. Visibility graphs replace the isovist as node content by mere intervisibility information translated into edges to other nodes that are now distributed on a regular and dense occupancy grid of possible observation points. This technique facilitates the derivation of global or second order measurands like for example on visual stability that may be relevant for locomotion and navigation. Indeed, recent empirical studies have shown that visibility graphs are useful to predict spatial behavior and affective qualities of indoor spaces.

As apparent from the previous section, formally similar graph-like representations of space serve different purposes in spatial cognition and in architecture. While graphs in spatial cognition are mainly used as models for mental representations of environments, in architecture they are used as generic formalized description-systems for the structure and shape of built environments. However, despite these different perspectives, the concept of environment (i.e. the sum of behaviorally relevant aspects of an organism's habitat) as represented content is another fundamental communality between the disciplines implicating further parallels. The combination of general compatibility and independent directions of development makes integrative concepts appear as particularly promising. (Franz, Mallot, & Wiener, 2005)

In order to study and analyze spatial behavior in architectural and urban design, graph based techniques are commonly used as representations of space. However, despite this fact, only weakly defined rules exist that describe how these place graphs are generated from the environment. Usually, they are hand-made by selecting all possible places (graph-nodes) as well as all the connections between these places (edges). While this method is applicable for simple environments consisting of clear-cut subspaces or for simple street grids, a more generic approach would be clearly favorable. The 'adaptable network' in this sense can be considered as a computational model which demands a more generic and multi layered analysis approach. The results of the studies that have been done on graph based analysis concerning the 'adaptable network' will be gathered in the conclusion section of 'spatial analysis in architecture' chapter.

2.5.4. Cellular Automata;

Cellular automata is a method of computing which can simulate the process of growth by recounting a complex system by simple individuals/characters following basic rules. This concept of simulating growth was brought out by John von Neumann and later involved into simulating multi-state machines by Ulam. The conception is much sought after by Martin Gardner when he specified John Horton Conway's 'Game of Life', simply known as 'Life'. The universe of the game that Gardner generated two-dimensional patterns from, is an infinite two-dimensional orthogonal grid of square cells. Stephen Wolfram launched a research of the conception to stand for the physical phenomena and recently re-introduced the review/compilation in 'A New Kind of Science’’. (Krawczyk, 2002)

Cellular automata has the ability to generate patterns. The results can be submitted as organized patterns to tender architectural forms. There is such a reference between cellular automata and architecture constitutes that, in the terms of this thesis, this can be assumed an encouraging source of search. Considering the current results which are the preliminary phases of future results, a cellular automaton differs from a traditional deterministic method, as it is considered as a mathematical approach. Until a certain accomplishment gained, this iterative replacement method continues. Fractals and strange attractors are also created in a similar manner. Many digital methods in architecture are parametrically driven, an initial set of parameters is used to generate one result. (Krawczyk, 2002) When the alternative is needed, the parameters should be modified so the generation is to be repeat anew. As the difference between these two methods is similar to the one in parametric methods, one can say, unlike in recursive methods, in parametric methods the results can be easily anticipated.

Possible architectural patterns can be developed from this rich and interesting platform offered by this context. This retrieval by reformulating the phases; Wolfram, one dimensional, Conway, two-dimensional, and Ulam, three-dimensional, leads to an evolvement for the universe of the cellular automata over a number of dimensions.

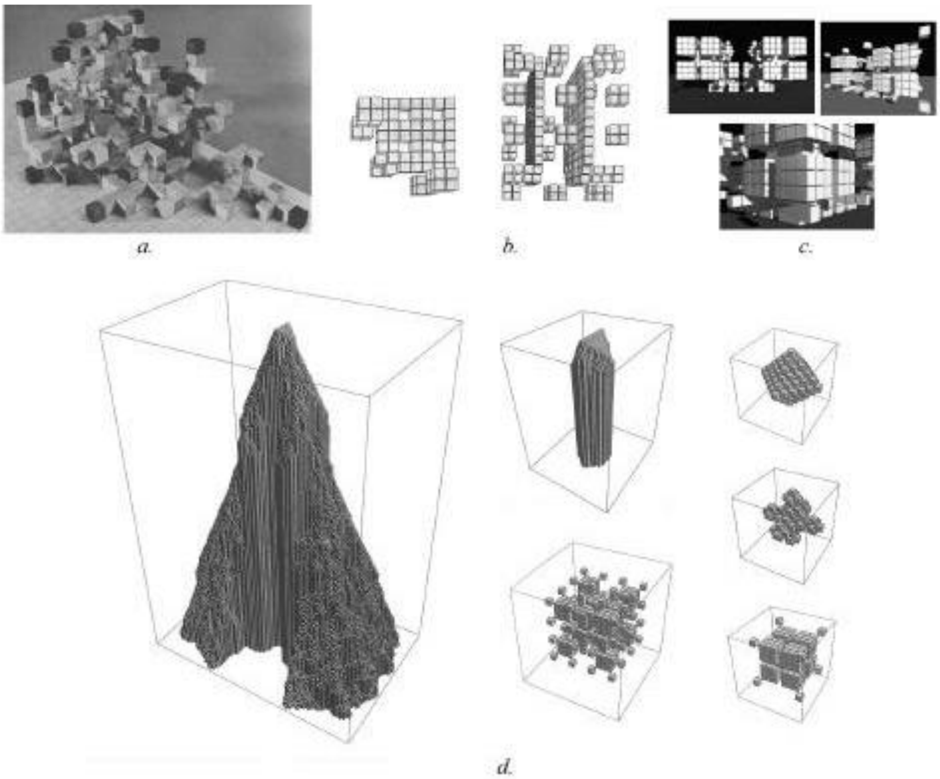


Figure 8:Three-dimensional cellular automata *(Krawczyk, 2002)*

The wooden block model, Figure 8a, created by Schrandt and Ulam, seen as an early example of three-dimensional pattern development. Figure 8b, inquired into repeating patterns as Conway studied in two dimensions is Bays; and Figure 8c, be said much in the same spirit, highly inspirational eventual architectural application by Coates. Even with the difference in the approach and in the application, the remarkable resemblance in two methods developed by Wolfram, explicit representations of the cellular automata is clear. Here, Figure 3d, in which a stacking method is explored, as similar to the Bays.

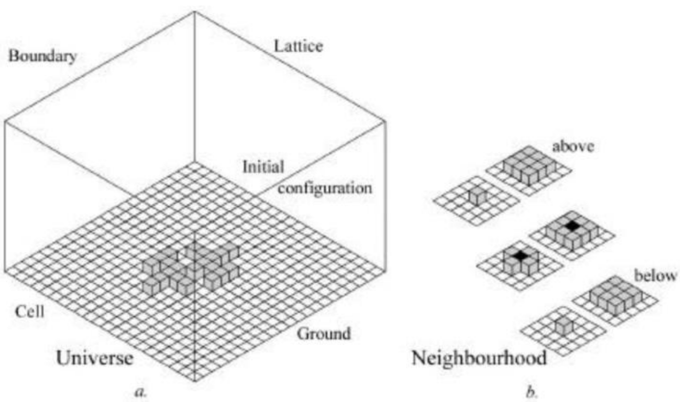


Figure 9: Basic cellular automata terminology (Krawczyk, 2002)

Considering each cell has a specific state (occupied or empty), and its location represented and recited by a marker, Figure 9a, is the three-dimensional universe of cellular automata, consist as an unlimited lattice of cells. In the light of the rules that determine a cell's future by taking account of the cell's neighborhood, weather it survives, dies or is born in the next generation, the transitional process begins with an initial state of occupied cells and progresses by a set of rules to each succeeding generation. Figure 9b evinces two common methods of determining which conterminous cells to consider, as neighborhood can be specified in a number of ways. By checking each occupied cells' neighborhood, the rules developed by Conway are;

1. Any live cell with two or three live neighbors live on to the next generation, while with fewer than two live neighbors, the cell dies, as if caused by under-population.
2. Any live cell with more than three live neighbors dies, as if by overcrowding.
3. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction. (Krawczyk, 2002)

Relying on the context of use and the objective used for, cellular automata can be considered as an analysis method or a pattern generator. On the subject of the proposed computational model cellular automata is not an implementation that can be used straightaway as a form or pattern generator while as a simulation agent, the fundamental principles can be followed. The principles adopted from cellular automata concerning the relational rule based growth and the die or live, can be used in two levels referring to the proposed model in this thesis. First the grid generation with respect to the boundaries and the obstacles in a site and second the incremental growth of organizational cells. Although cellular automata provides important 3D features as well, both of the adopted features are focusing the 2D features of the method.

This chapter contains the topics studied in relation to the aims of the proposed model. The position of the arguments that are proposed through the developed computational model are studied and investigated in this chapter. After investigating the topics studied above of generative design, network models, temporary space organizations and spatial analysis methods, the internalization and implementation of these knowledge is presented in the following chapter. Next chapter illustrates the theoretical background of the computational model proposed in this thesis with fundamental features and necessary formulations.

CHAPTER 3

THEORETICAL BACKGROUND OF ANG

The advanced use of computer assists creative production in many different ways depending on the goal of the designer. However, one of the most crucial elements of any architectural design process; the analysis phase cannot equally benefit the current technological advancements comparing to other uses of computational tools in the field of design. The results of many analytical studies in the pre-design phase are generally dominated by the intuitive contributions of the designers. This means that the results are interpreted instead of being integrated to the computational process. In the scope of this thesis, a combination of a multi-layered analysis tool and a generative system has been proposed in order to build up a better integration.

This chapter presents the mathematical background of the computational model proposed in this thesis, namely ANG (Adaptable Network Generator) with necessary formulations and definitions. In this chapter the definition of the ANG, the objectives of ANG are explained and it is followed by an introduction to the structure of the proposed system, showing the data flow. The inputs and the outputs of the ANG are explained and visualized with the help of figures and diagrams.

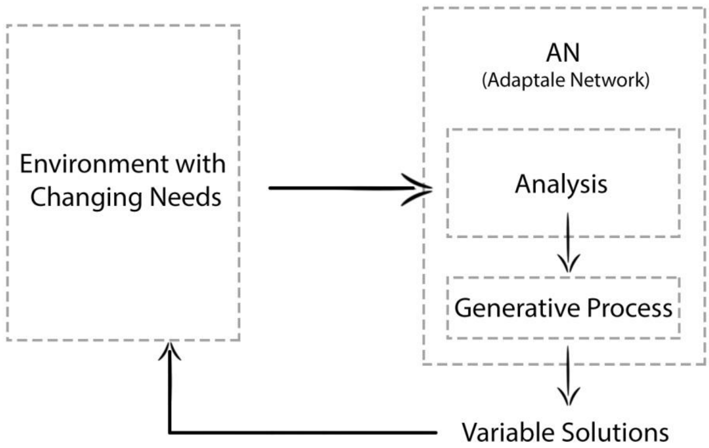


Figure 10:Conceptual Diagram Showing the Relations Between ANG and the Context

3.1.Definition of the ANG (Adaptable Network Generator)

ANG is a generative system proposal for supporting architects in the early design phases. For a given site and a given context ANG generates circulation paths, cellular organizations and connections as networks. These can be used to provide circulation options for users, organizational schema for temporary space design and optimized connections for infrastructure.

3.2. The Objectives of ANG

One of the primary reasons for developing such a computational model is to help designers to control the complexity, specifically when they are working in contexts with changing needs. Cases like temporary space design are the main focus of this thesis and the primary application area of the ANG. The other objectives for the ANG are as follows:

a. To mark the obstacles on a site which keep the designer from placing objects
b. To simplify and structure the complexity of the topography by mapping a grid
c. To convert the site into a multi cellular order
d. To help the designer for making easy adjustments on the multi cellular grid
e. To isolate occupied cells from active cells
f. To generate 'Shortest Walk' options
g. To generate minimum cost/time path options
h. To supply the Average Distance data from one cell to all other active cells
i. To provide 'Breadth-First Search' data in order to understand contiguity
j. To provide 'Betweenness Centrality' data in order to understand the centrality

The proposed computational model is capable of combining the analyzed data with the generative tools in a parametric medium. In other words ANG provides a virtual ground of analytical data where designers can generate possible spatial organizations without losing the connection to the current projects environment and its context. The reason this computational model is called 'adaptable network generator' is because it functions as a network generator with an aim to create reciprocal relationships between the variable solutions and the dynamic flows and fluxes of the environment.

The steps summarized above are studied in detail in the following sections under two main layers that form ANG.

ANG contains two fundamental layers;

1. Spatial Analysis Layer

2. Generative Layer

3.3. Spatial Analysis Layer

Spatial analysis layer is mainly designed to analyze the dynamic and static characteristics of environments with changing needs. Spatial Analysis Layer consists of more than one analysis method which makes it a multi-layered model. These methods are all combined under the Spatial Analysis Layer in order to supply the resulting data of the conducted analysis to the Generative Layer.

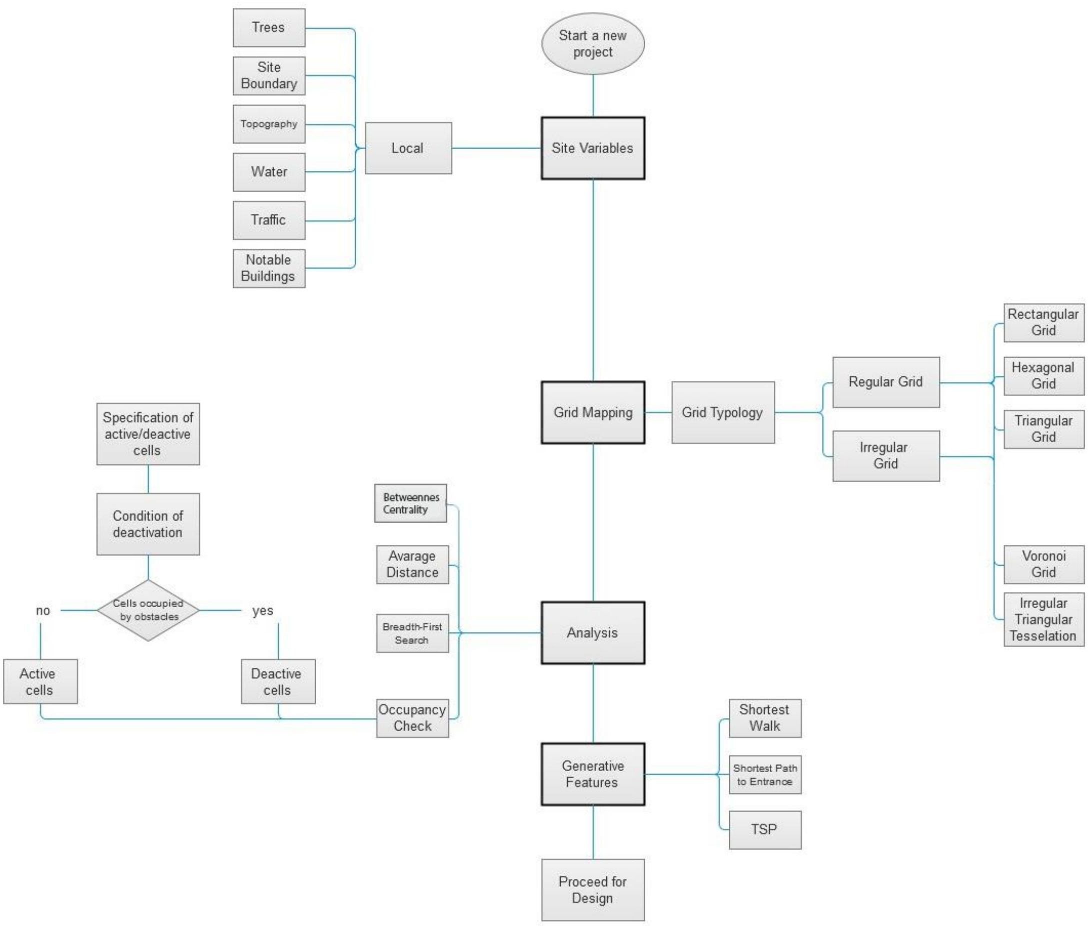


Figure 11: Conceptual Diagram of the ANG (drawn by the author)

The main purpose of this layer is to create reciprocal links between the components of a system, the system as a whole and the environment. These links are formed in such a way that every link contains a set of information derived by the spatial analysis tools. In other words it is a computational agent that can perform analytic operations for generating adaptable design solutions and for conducting analytical readings over the generated results.

Figure 11 shows the studied spatial analysis techniques for the development of the multi-layered analysis method. As a result of the studied techniques in chapter 2, the relevant methods concerning the ANG are; Occupancy Grid, Place Graph, Access Graph, Axial Maps and Cellular Automata. The adaptation and the internalization of the knowledge learned from these techniques require a critical interpretation and transformation. Therefore the steps performed by the resultant multi-layered spatial analysis method are formalized as follows;

1. Cellular Mapping;
    a. Grid Selection
    b. Grid Implementation
    c. Grid Adjustment
2. Obstacle Elimination
    a. Active/Unoccupied Cells
    b. De-active/Occupied Cells
3. Shortest Path Analysis
    a. SP (Shortest Path)
    b. TSP (traveling salesman problem)
4. Average Distance Analysis
5. Breadth-first Search
6. Betweenness Centrality Analysis

One important point should be noted that the steps listed above including cellular mapping, obstacle elimination and shortest path analysis are developed in the scope of this thesis an can be interpreted as a direct contribution to the field. On the other hand the steps including average distance, breadth-first search, betweenness centrality analysis which are very important steps of the ANG, are implementations of existing algorithms. The tools that are used to perform these analysis belongs to an add-on called SpiderWeb for Grasshopper, developed by a group called GBL[1] from Vienna. SpiderWeb is used inside ANG in order to the do the analytical reading of the results generated by SP component and to have analytical information over the whole site before the generative stage if desired.

3.3.1. Cellular Mapping;

Cellular mapping is the first step where the required adjustments are performed in order to create stable base for ANG to work on. All the analytical and generative features of the ANG are built upon the implemented grid on the selected site. Cellular mapping consists of three steps; Grid Selection, Grid Implementation and Grid Adjustment.

As one of the oldest architectural design tools, the grid is an efficient and useful tool for controlling the position of the building elements (Gross). The reason why a grid is used as a base for ANG is that grids can help designers control the positions of the space elements by making the layout more systematic. On the other hand, by determining positions of different building elements in relation to a grid or to a set of grids, specifying design rules and computational rules become easier to control and more efficient.

Grid selection is the first stage of the cellular mapping process. In order to generate a well-defined grid depending on the event related characteristics of the site, the designer should determine a suitable grid typology for the implementation. The set of possible grid typologies consist of two main categories; Regular Grids and Irregular Grids.
With respect to the limits of ANG, the possible regular grids that can be used are; rectangular grid, hexagonal grid and triangular grid. Different types of regular grids can also be used but there is a possibility of failure in the process. The grid typologies mentioned above are tested and works with ANG without any errors.

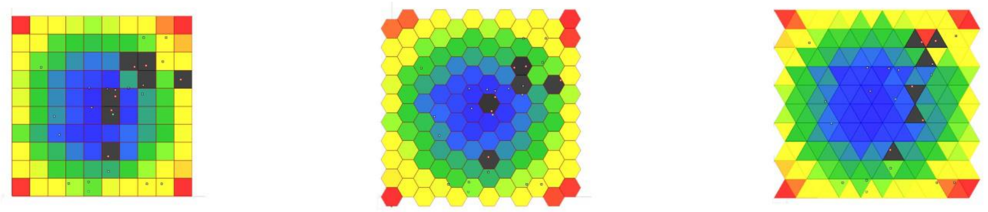


Figure 12: Regular grid typologies. Rectangular grid (left), hexagonal grid (middle), triangular grid (right). (drawn by the author)

[1] http://www.gbl.tuwien.ac.at/Archiv/digital.html?name=SpiderWeb

For generating the base of ANG irregular grids can also be preferred. Irregular grids represent not only grids but also the irregular tessellations and shape packing solutions like; Voronio grid, irregular triangular tessellation.

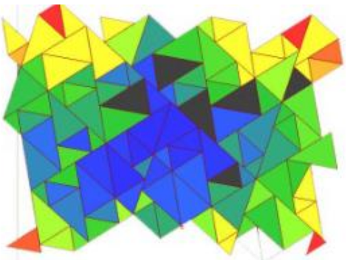


Figure 13:Irregular Triangular Tessellation (drawn by the author)

Second stage is called grid implementation. In this stage, the designer is expected to prepare any required data about the site. (ex. site boundaries, existing objects in the site etc.) This process is done by manual use of computer by the designer. The main purpose of the grid implementation is to fit the grid into the site boundaries so that all the adjustments following this stage will be executed inside the boundaries of the project site.

Grid Adjustment is the final stage of the cellular mapping. In this phase the designer can adjust the cellular structure of the implemented grid. These adjustments can be done according to site variables or event related variables such as; user count, event type, total area required etc. Generally these variables affecting the grid topography are determined by the designer but also many design decisions in the late design processes are also linked to the very first stages of the system such as cellular mapping. In other words feedback loops that can be possibly created in the post design stages of ANG are always linked to the grid adjustment stage.

3.3.2. Obstacle Elimination;

Obstacles in a site are the representations of objects that occupy a cell of the implemented grid. They are represented as points or set of points. The cells of the grid are divided into two depending on the condition of being occupied or not occupied by these points. The occupied cells are called de-active cells which mean that it is not possible to place a temporary space design on that location. On the other hand unoccupied cells are called active cells signifying that these locations are free to build.

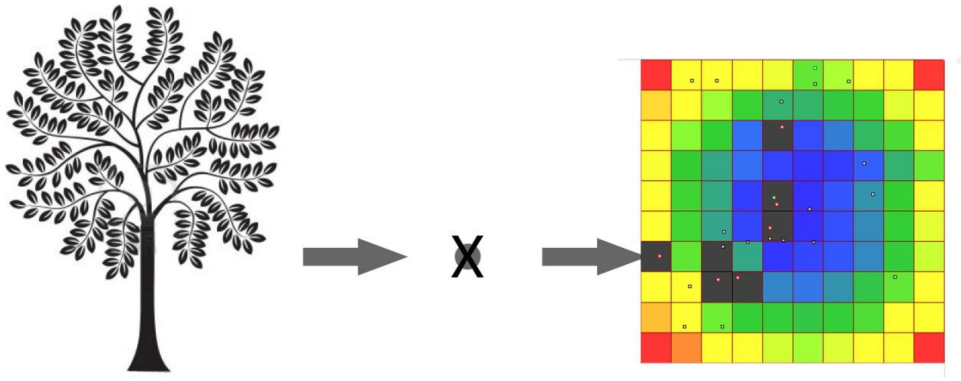


Figure 14: Obstacle Detection (occupied/unoccupied cells) (drawn by the author)

3.3.3. SP (Shortest Path);

SP component is developed as a component for Grasshopper to find the shortest path between two given points on a cellular organization to create multi-cellular temporary design options. There are two main reasons to develop such a tool.

First, the similar examples in the market generally present the best result which is just 'one' of the possible results. In the scope of this thesis where the entire computational model is developed on a user-defined grid typology, potential of having variety of possible results is increased. Besides from a designer point of view a selection made from a set of results is always more satisfying and gives the designer the chance to compare the results. Since the results may be mathematically equal, they may not be satisfactory regarding aesthetics and other qualities. Therefore a shortest path component that is capable of producing all the possible shortest-paths has been developed.

Second, the existing shortest-path algorithms generally search the shortest path inside a point cloud. On the other hand the SP component developed in this thesis works with closed polygons. Which means instead of generating a linear shortest path from point A to point B (Figure 28), SP generates a set of closed polygons (cells of a grid) that are adjacent to each other following the shortest possible paths (Figure 16).

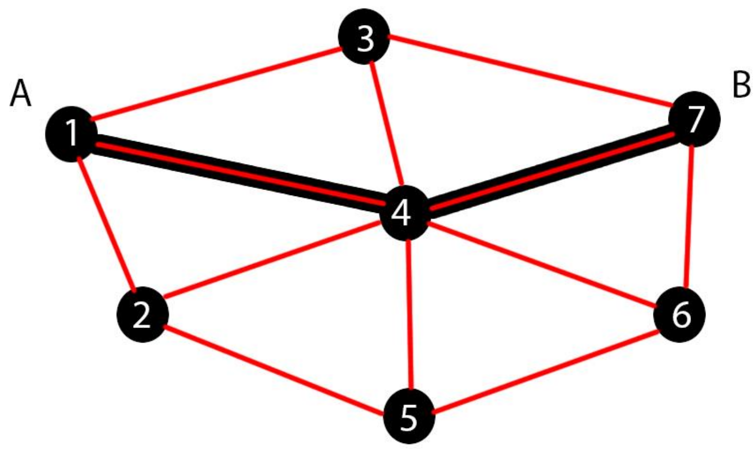


Figure 15: Result of a standard shortest path algorithm (drawn by the author)

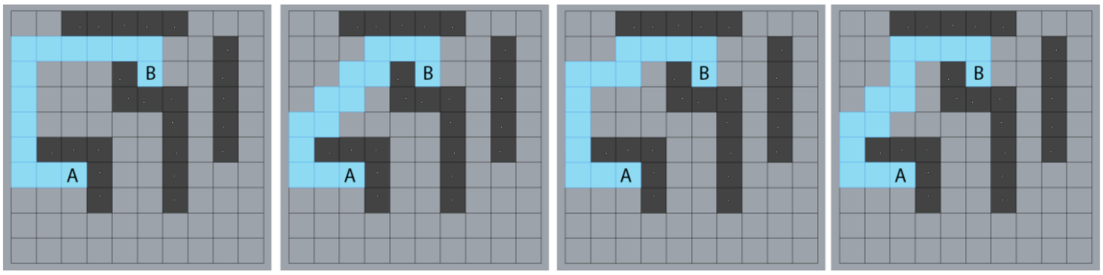


Figure 16: Possible results of SP component. Dark cells are the deactive/occupied cells and the blue cells are the generated shortest paths (drawn by the author)

In order to have a better understanding on the SP component and its ability to generate shortest path options the 'shortest path problem' is studied.

3.3.3.1. Shortest Path Problem;

Optimizing the distance between two predefined points has always been a desirable subject for many applications. The shortest path problem investigates the collection of paths that comprises the shortest path from a specified node s, called the source, to a second specified node v, called the destination. Figure 30 shows a typical network where s = 1 and v = 5. The length is represented by the parameter enclosed in parenthesis on the paths. Each arc is numbered sequentially. (Hübner & Mallot, 2002)

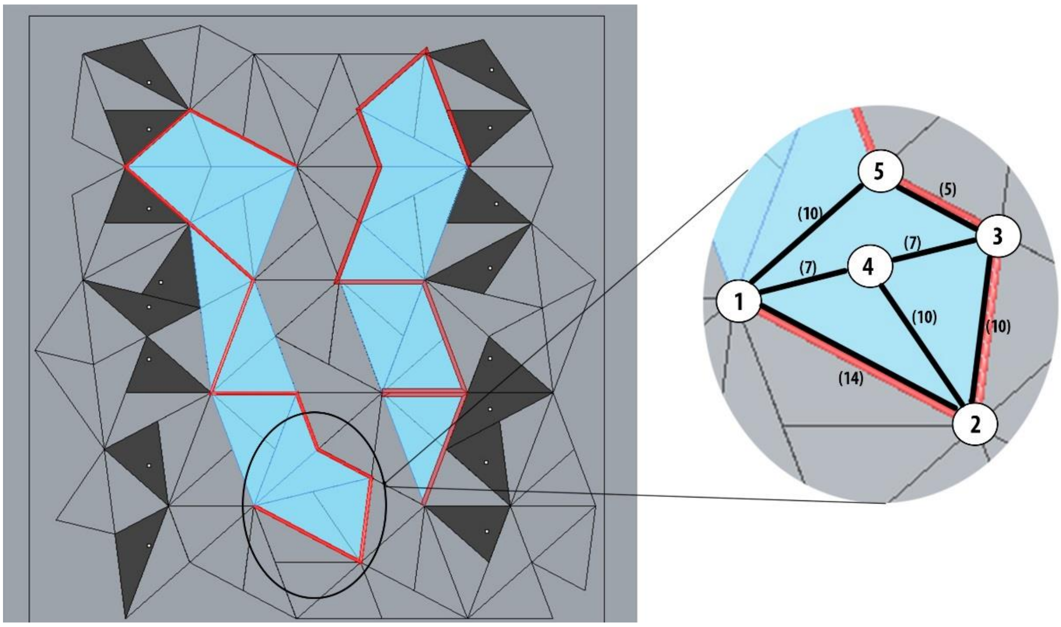


Figure 17: Graph Network Showing Path Lengths (drawn by the author)

The algorithms that are widely in use to solve shortest path problem are called Dijkstra and A* (A star) algorithms. During the development of TSP and SP, Dijkstra and A* algorithms are studied. Dijkstra and A* algorithms both have efficiencies depending on their applications. Both of the computational tools developed in the scope of this thesis (TSP and SP) use Dijkstra algorithm. The basic reason behind it is that A* makes searches from one cell to the next one with a heuristic approach. On the other hand Dijkstra searches from all points to all points using a genetic approach which suits to our objectives because we are looking for a path that visits all active cells with a minimum detour. That is why Dijkstra algorithm is studied in detail.

3.3.3.2. Dijkstra's algorithm;

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. (Dijkstra, 1959)

The algorithm finds the path with lowest cost (i.e. the shortest path) for a given source of vertex (node) in the graph. It searches between the source vertex and every other given vertex. Another application of this algorithm is finding costs of shortest paths from a single vertex to a single destination vertex. The algorithm stops when the shortest path to destination has been determined. This shortest path algorithm is widely used in network routing protocols, most notably IS-IS (Intermediate System to Intermediate System) an OSPF (Open Shortest Path First). For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. In the case of ANG this algorithm finds the shortest path between the source cell of the grid and the destination cell of the grid. (Dijkstra, 1959)

3.3.3.3. Formulation of Dijkstra Algorithm in SP Component;

The Dijkstra algorithm applied in the SP component that is developed in this thesis is formulated as follows;

Graph is a collection of vertices and edges; G = (V, E) and every edge has a length. In other words the length (L) of an edge in a graph is the distance between u and v assuming L (u,v) ≥ 0. For example L (u,v)=5. If there is no edges directly from u to v it is assumed that the length or distance is ∞. dist (u,v) = ∞. For a node u∈G, dist (u) is defined as the distance of the shortest-path from s (starting vertice that is already in the shortest path set S) to u.

For the next node to create an edge the algorithm searches for the shortest-path from the set of V-S (vertices that are not in the set of S). d'(v)=min[dist (u)+L(u,v)] where *u*∈*S*.

S={s}, d(s) = 0, $\tilde{E} = \emptyset$
while *S*≠*V* ,
Select a node to
minimize [d(*u*)+*L*(*u*,*v*)]
u∈*S*
add *v* to *S*
and add (*v*,*u*) to , $\tilde{E}$
end while

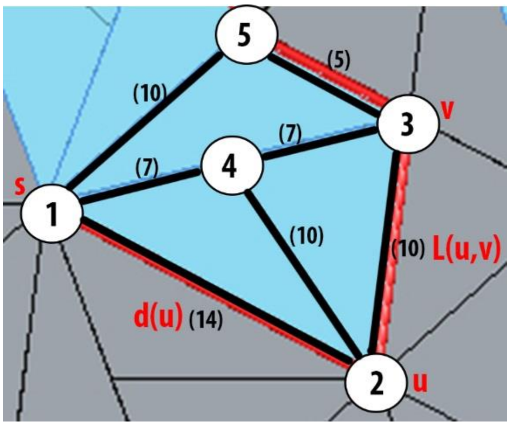


Figure 18:Formulation of shortest-path calculation on irregular triangular grid (drawn by the author)

This is the basic formulation of Dijkstra algorithm to calculate shortest-paths. The contribution that has been done in this thesis by developing the components of Shortest-path and TSP is the way this algorithm is applied to cellular organization to generate shortest-path options for temporal space organizations. The formulation goes as follows;

ANG works with closed polygons instead of working directly with point clouds. The process starts with inputting the generated grid as a mesh which is technically a set of closed polygons (the cells that forms the grid). The shortest-path component decomposes the closed polygons into vertices/nodes. In a standard shortest-path calculation like the one formulated above, the algorithm generates possible graph edges and search for the minimum sum of the edge distances. In the case of ANG the algorithm starts with a condition of adjacency which means that before the shortest-path calculation ANG checks if the grid cells are neighbors or not.

for dist (u,v)
if the number of common points between two polygons = 2 (which means they are adjacent)
dist (u,v) = Euclidean distance
V∈N   N is the set of centroids of the adjacent polygons
if the number of common points between two polygons =1 or 0
dist (u,v) = ∞
V∉N

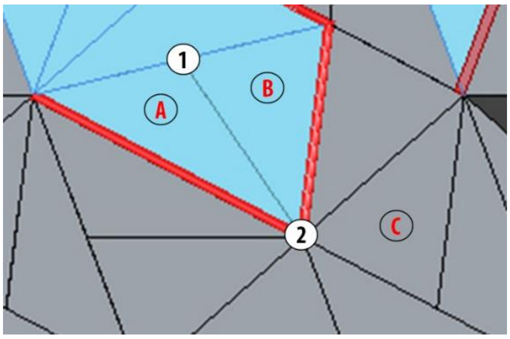


Figure 19: Showing adjacency conditions between grid cells (drawn by the author)

In Figure 19 polygon A and polygon B shares 2 common points so they fit the adjacency condition and their centroids are taken into account for shortest-path calculation. On the other hand polygon C shares only one common point with A and B so the distance to centroid C is equals to ∞ which means the algorithm does not see C as a neighbor and leaves it out of the calculation.

After adjacency condition is met the algorithm searches for the shortest distance between the center point of the starting polygon and the center point of the neighboring polygons. Therefore the formulation of the Dijkstra becomes this;

$S=\{s\}$, d(s) = 0, $\tilde{E} = \emptyset$
while $S \neq V$,
Select a node $v \notin S$ to
minimize [d(u)+L(u,v)]
$u \in S$ $v \in N$
add v to S
and add (v,u) to $\tilde{E}$
end while

SP component designed in a way to minimize the multifunction and to increase the control of the designer on the component to make adjustments on the multi-cellular shortest path that will be generated. In order to do that two additions were made to the algorithm that is described above namely; the Tolerance and Threshold. These additions aim to minimize the complications that occur when working with irregular grids.

3.3.3.4. Tolerance Condition;

'Tolerance' defines the range of closeness of two points of an edge in order to assume and operate them as one point. In other words it defines range of two points to be accepted as overlapping points. This situation happens in irregular grids because in the generation of irregular grids very small edges can occur which are not visible to human eye. The elimination of the overlapping points prevents the system to create unwanted adjacency conditions.

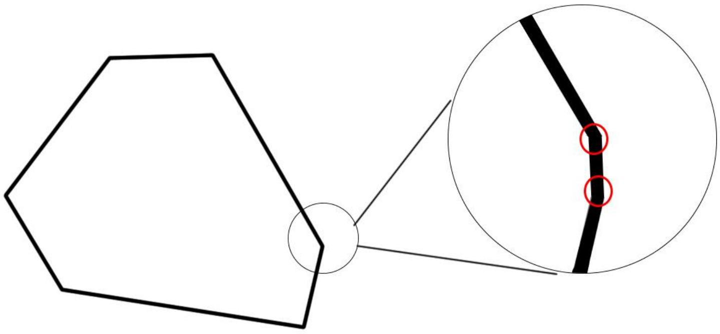


Figure 20: Diagram showing the two very close points that are forming a very small edge that are accepted as overlapping points because they are in the range of the tolerance (drawn by the author)

3.3.3.5. Threshold Condition;

Second addition to the existing algorithm is the 'Threshold'. Threshold defines the minimum length of a common edge between two polygons in order to be accepted as neighbors. This gives the designer to control the outcome of the component by adjusting the adjacency condition. In other words in order to be neighbors a polygon should meet not only the rule of having two common points on an edge but also needs to meet the threshold condition set by the designer.

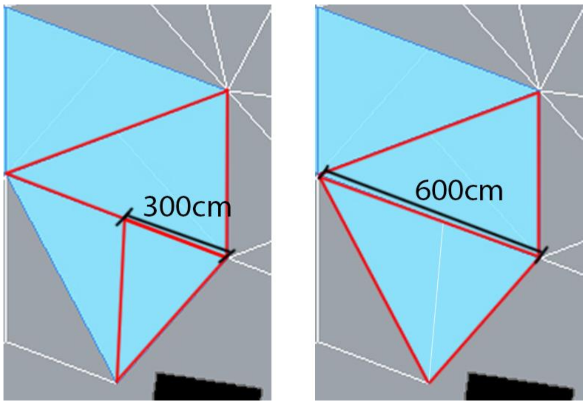


Figure 21: Diagram showing two different results of the Threshold condition
Left one; threshold is lower than 300
Right one; threshold is larger than 300 (drawn by the author)

3.3.4. TSP (Traveling Salesman Problem);

The TSP is extensively studied in literature and has attracted since a long time a considerable amount of research effort. Basically, the TSP is the problem of a salesman who wants to find, starting from his home town, a shortest possible trip through a given set of customer cities and to return to its home town.

Traveling salesman problem can be modeled as an undirected weighted graph. In such a graph model cities are the graph vertices, paths are graph edges and path distances are the edge lengths. It is basically a minimization problem starting and finishing at a specified vertex after visiting each of the existing vertices exactly once. In the case of ANG, graph nodes are vertices of the grid cells, the graph edges are paths and the lengths of the edges are the distances from one vertice of the cell to the other.

During the development of TSP Dijkstra algorithm and genetic algorithms are used. The formulation of the TSP component goes as follows;

TSP component starts with creating a matrix of all vertices supplied by the designer. The distances from one node to all other nodes are stored in this matrix. But the networks that are studied in this thesis are not fully connected networks so a contiguity condition is set.

Contiguity Condition;

for dist(u,v)

if the number of common points between two edges = 1 (which means they are connected) dist(u,v) = Euclidean distance

$V \in N$ N is the set of vertices of all the edges in the network

if the number of common points between two polygons = 0

dist(u,v) = ∞

$V \notin N$

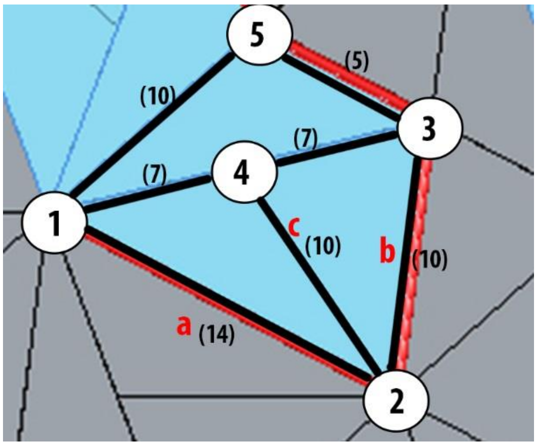


Figure 22: Showing contiguity conditions between the edges of polygons (drawn by the author)

In Figure 22 the contiguity conditions between the edges of polygons are represented. Edge 'a' is connected with edge 'b' through their common point 2. On the other hand if the algorithm wants to go from point 2 to point 5 although there is no direct edge connecting them together in order not to set the distance to ∞ TSP runs Dijkstra algorithm and finds the shortest path between those points and uses the distance of that path.

In other words if the contiguity condition is not met, in order to calculate the distance between these unconnected nodes Dijkstra algorithm is used to generate the shortest path between these two nodes so that the distance is stored in the matrix as a numeric (Euclidean distance) value instead of ∞.

if dist(u,v) = ∞

$S=\{s\}$, d(s) = 0, $\tilde{E} = \emptyset$

while S≠V ,

Select a node $v \notin S$ to

minimize [d(u)+L(u,v)]

$u \in S$

add v to S

and add (v,u) to $\tilde{E}$

end while

When the matrix is created with all the distances stored inside, the genetic algorithm starts to calculate and search for the shortest detour.

3.3.4.1. Outline of the Basic Algorithm;

In order to understand the implementation of the basic algorithm to the ANG a standard genetic algorithm is studied. A basic genetic algorithm starts its calculations by generating a random population of *n* chromosomes (suitable solution for the problem). The second step is called the fitness function. In this step the algorithm evaluates the fitness *f(x)* of each chromosome *x* in the population. Then it creates a new population by repeating the following steps until the new population is complete (Obitko, 2014).

- ***Selection*** *Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)*
- ***Crossover*** *With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.*
- ***Mutation*** *With a mutation probability mutate new offspring at each locus (position in chromosome).*
- ***Accepting*** *Place new offspring in a new population*

Before the algorithm loops and goes back to the second step it replace the offspring with a new generated population for a further run of the algorithm and tests if the end condition is satisfied. If not it stops and returns to the best solution in current population (Obitko, 2014).
Genetic algorithm that is used in TSP component follows a very common list of steps namely;

- Encoding
- Fitness
- Crossover
- Mutation
- Selection

3.3.4.1.1. Encoding;

The encoding that is used in TSP component is called Permutation Encoding. Permutation encoding is generally used in ordering problems. Traveling salesman problem and task ordering problem are two very common uses of permutation encoding (Obitko, 2014)
In permutation encoding, every chromosome is represented as a sequence of numbers. For example;

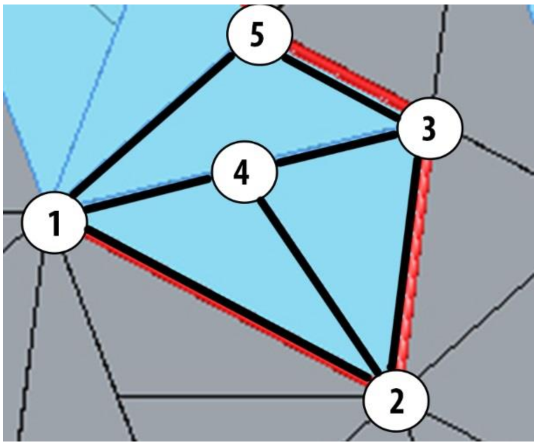


Figure 23: Source nodes for permutation encoding (drawn by the author)

Chromosome A - 1/5/3/2/4

Chromosome B - 5/2/3/4/1

In TSP a chromosome represents the order of vertices, in which the user will visit them.

3.3.4.1.2 Fitness;

A value for fitness is assigned to each solution (chromosome) depending on how close it actually is to solving the problem. (Boukreev, 2014) In TSP component the fitness function f(x) is the minimum length of the summation of the distances between visited nodes.

3.3.4.1.3. Crossover;

For crossover operation the Greedy Crossover method by J. Grefenstette is selected. "Greedy crossover selects the first city of one parent, compares the cities leaving that city in both parents, and chooses the closer one to extend the tour. If one city has already appeared I n the tour, we choose the other city. If both cities have already appeared, we randomly select a non-selected city." (Louis, 2010)

3.3.4.1.4 Mutation;

In TSP component mutation method is set as the greedy mutation.

"The basic idea of greedy-swap is to randomly select two cities from one chromosome and swap them if the new (swapped) tour length is shorter than the old one" (Louis, 2010)

3.3.4.1.5 Selection;

For selection operation Roulette Wheel Selection is applied. In roulette wheel selection, parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. (Obitko, 2014)

The algorithm is codded by using programing language C# and implemented in the parametric model developed in Grasshopper. It first creates 100 random permutations containing all the vertices once in the encoding. These permutation encodings are tested by a fitness function. Fitness function searches for the minimum total distance of the paths defined in the permutations. By doing crossovers between the fittest elements of the paths, new mutations are generated. In TSP component this process is repeated for 50 generations in order to keep the calculation time and the computing power limited. Finally from the best generation the fittest path is selected.

TSP component does not always give the best solution because it starts with a random initial state. That is why a seed input is added to the component to give the designer the chance to re-calculate the result if it is unsatisfactory.

3.3.5. Average Distance Analysis;

The distance between two vertices of a given graph is a quite simple but unexpectedly useful notion. Several graph parameters such as the diameter, the radius, the average distance and the metric dimension are derived from it. The average or mean distance between two vertices is the measure of graph compactness. It has been used in the evaluation of the architectural floor plans. (March & Steadman, 1974)

In ANG average distance is one of the analyses that have been used in analysis layer. The purpose of this step is to have a general understanding and metric data about the connectedness, betweenness, closeness or farness of a selected cell to all other cell in a spatial organization. The results of such an approach may inform the designer about where to place the most central function or help them to understand the importance of the selected location relative to other possible locations. On the other hand this data is also integrated to the SP component that is developed in this thesis in order to do analytical readings of the multi-cellular organizations with respect to average distance analysis.

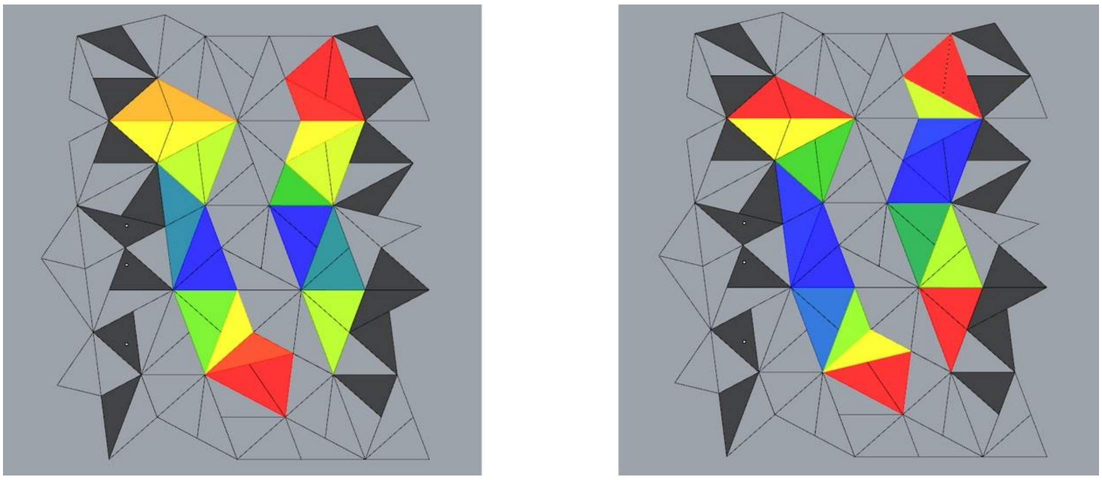


Figure 24:Average distance analysis from an entrance in the center to all other cells (Left). Average distance analysis from every cell to all other cells (Right) Hue represents the average distance (blue=min and red =max) (drawn by the author)

Figure 24 illustrates two generated multicellular organizations and their average distance analysis. The image on the left shows the average distance of every cell to the entrance on the other hand one can read the average distance data from every cell to all other cells from the image on the right. The image on the right gives a general understanding on the overall distribution of the individual cells in a multi-cellular organization.

A path in a graph is a set of distinct vertices in a sequence, in other words the adjacent vertices in a sequence is also adjacent in the graph. Assuming that all the weights are nonnegative and all the graphs are connected the definition of a path length can be made as follows. For an unweighted graph, the length of a path is the number of edges on the path. For an (edge) weighted graph, the length of a path is the sum of the weights of the edges on the path. Assuming the conditions above are met the formulation is done as follows. The distance d(u,v) between two vertices u and v of a finite graph is the minimum length of the paths connecting them. If no such path exists (i.e., if the vertices lie in different connected components), then the distance is set equal to ∞. In a grid graph the distance between two vertices is the sum of the "vertical" and the "horizontal" distances.

In Figure 25, average distance analysis in different generic grid typologies are represented. Three regular generic grids including; rectangular grid, hexagonal grid and triangular grid are analyzed in addition to an irregular triangular grid. The results are generated on generic grids instead of multi-cellular organizations generated by ANG. Similar analysis can be performed in early design stages (before generating any organizations or designs) in order to perform analytical readings about the whole site.

In that sense Figure 25 shows that ANG is capable of working with different types of generic grid typologies with respect to decisions made by the designer.

In order to understand the theoretical background of the objectives (connectedness, betweenness, closeness or farness) of the average distance analysis used in ANG the topic of ‘centrality in a graph’ is studied.

3.3.6. Centrality in a Graph;

The importance of a vertex in a graph relative to the rest of the graph is determined by various types of measures of centrality with respect to the graph theory and network analysis. (e.g. how influential a person is within a social network, or, in the theory of space syntax, how important a room is within a building or how well-used a road is within an urban network). Many of the centrality concepts were first developed in social network analysis, and many of the terms used to measure centrality reflect their sociological origin. (Newman M. , 2010)

There are four measures of centrality that are widely used in network analysis: degree centrality, betweenness, closeness, and eigenvector centrality. For a review as well as generalizations to weighted networks, see Opsahl et al. (Opsahl, Agneessens, & Skvoretz, 2010) The topics that are related to the interests of this thesis are; Degree, betweenness and closeness centrality which are briefly explained and formalized below.

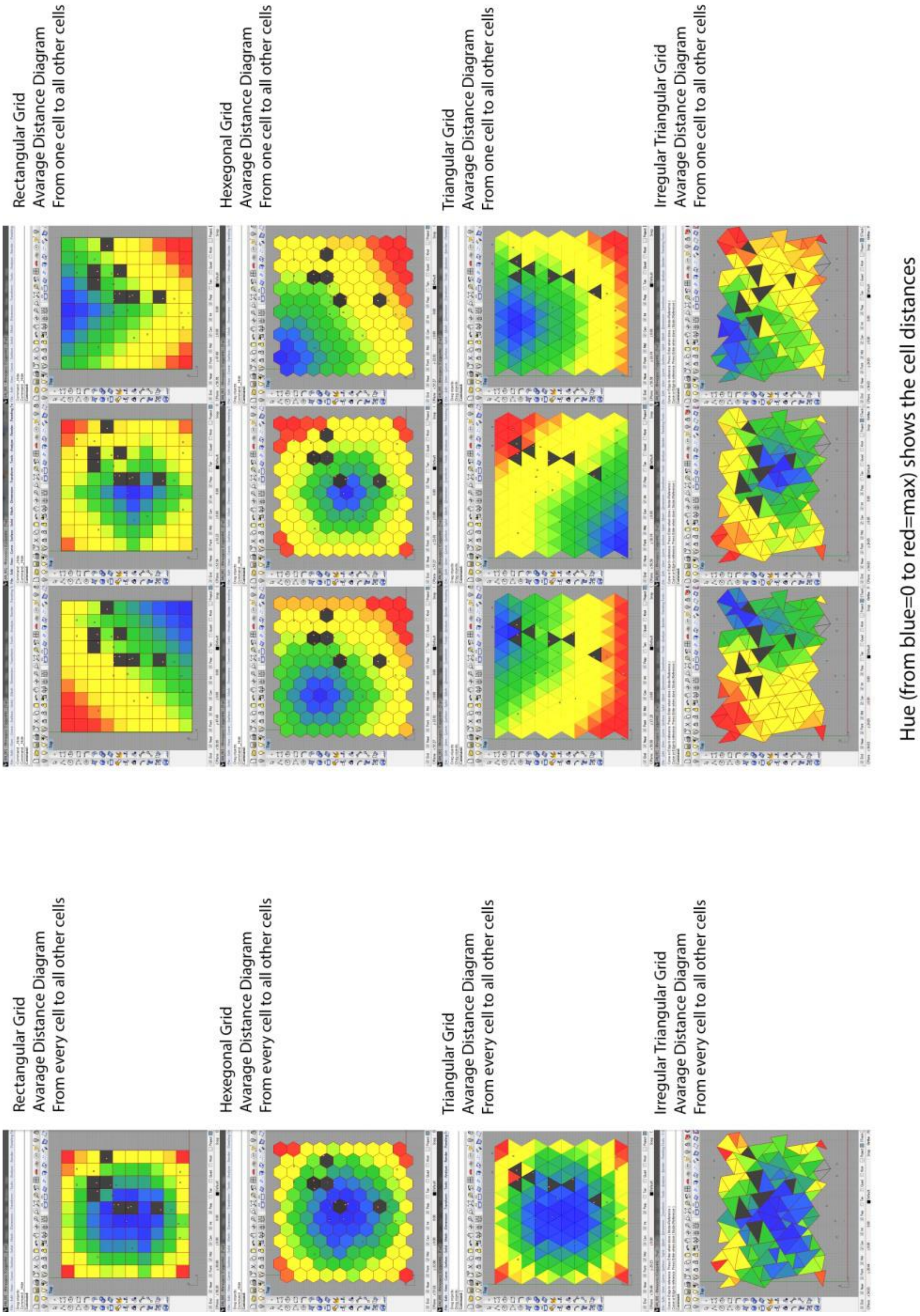


Figure 25: Average Distance Analysis in different generic grid typologies (drawn by the author)

3.3.6.1 Degree Centrality;

Degree centrality is conceptually simplest and historically the first term which is defined as the number of links connected to a node. (i.e., the number of ties that a node has). It can be calculated with the formula;

$C_D(v) = deg(v)$

Where the degree centrality of a vertex is *v*, for a given graph *G:=(V,E)* with |*V*| vertices and |*E*| edges.

3.3.6.2 Betweenness Centrality;

Betweenness is a measure of centrality of a vertex within a graph. Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It was introduced by Linton Freeman as a measure to quantify the control of human on the communication between other humans in a social network. (Freeman, 1977)

The betweenness of a vertex is calculated as follows;
The betweenness of a vertex *v* in a graph *G:=(V,E)* with *V* vertices is computed as follows:

1. For each pair of vertices (s,t), compute the shortest paths between them.
2. For each pair of vertices (s,t), determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).
3. Sum this fraction over all pairs of vertices (s,t).

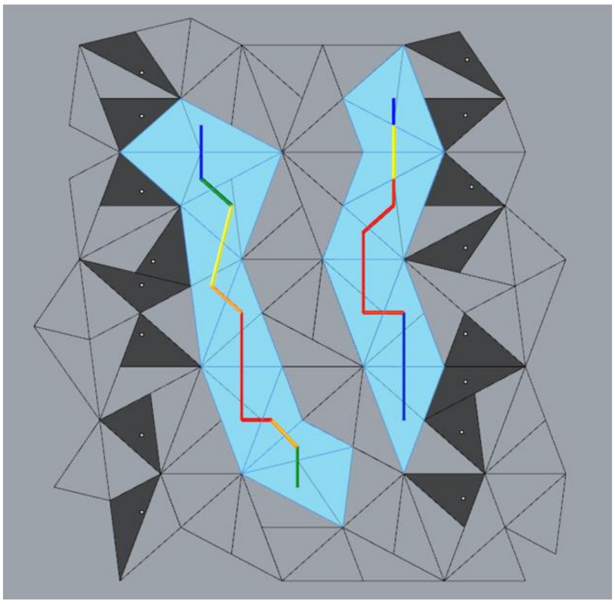


Figure 26: Betweenness Centrality of a multi-cellular design (red means max. centrality and blue means min. centrality) *(drawn by the author)*

3.3.6.3. Closeness Centrality;

In connected graphs the length of the shortest paths between all pairs of nodes defines the natural metric distance. The farness of a node s is defined as the sum of its distances to all other nodes, and the closeness is defined as the inverse of the farness. (Sabidussi, 1966) That is why the more central a node is the lower its total distance to all other nodes. Closeness can be regarded as a measure of how long it will take to reach from s to all other nodes sequentially. (Newman M. , 2005)

In the case of ANG the centrality conditions defined above can be read from the average distance analysis by studying the diagrams. If necessary these centrality measures can also be calculated using the above given formulations and the metric results of the average distance analysis. ANG also contains betweenness centrality analysis as a tool integrated to the proposed model. The result of average distance analysis in generic grid typologies is represented in Figure 25.

3.3.7 Breadth-first Search (BFS);

In graph theory breadth-first search (BFS) is a search operation that inspects all neighboring nodes from a selected node. The way BFS works is uninformed which means that the search method aims to expand and examine all nodes of a graph or combination of sequences by systematically searching through every solution. In other words, it searches the entire graph or sequence without a leading information, going through all combinations one by one until it finds it. In technical terms it does not use a heuristic algorithm. (Knuth, 1997)

The BFS is designed as a component for Grasshopper by GBL (Spider Web, 2014) the developers of the Spider Web plug-in. BFS is used as an analytical tool in the scope of this thesis in order to have two main results; metric and visual results. The list of metric results represents the distances between the neighboring cells if there are any. The visual result is a conceptual color diagram of the given organizational schema. In this diagram the equally distanced cells are represented in the same color where the others cells change color accordingly with the breadth.

3.3.7.1. The Algorithm of BFS;

The strategy of searching in a graph using BFS is limited to two fundamental operations;

a. Visit and inspect a node of a graph
b. gain access to visit the nodes that neighbor the currently visited node

The BFS starts with a root node and runs the search through all the neighboring nodes. Then for each visited neighboring node it restarts the inspection for their neighbors which were unvisited and so on.

The algorithm uses a queue data structure to store intermediate results as it traverses the graph, as follows:

1. Enqueue the root node
2. Dequeue a node and examine it
- If the element sought is found in this node, quit the search and return a result.
- Otherwise enqueue any successors (the direct child nodes) that have not yet been discovered.
3. If the queue is empty, every node on the graph has been examined – quit the search and return "not found".
4. If the queue is not empty, repeat from Step 2 (Knuth, 1997).

In Figure 27 results of the breath-first search analysis are represented. It is a generic case of a rectangular grid where the black cells are representing the occupied locations. Those cells are removed from the active cells list so that the algorithm generates shortest-path solutions from the unoccupied cells. In Figure 27 seven different shortest-path options are represented and analyzed in three different ways by the breath-first search component.

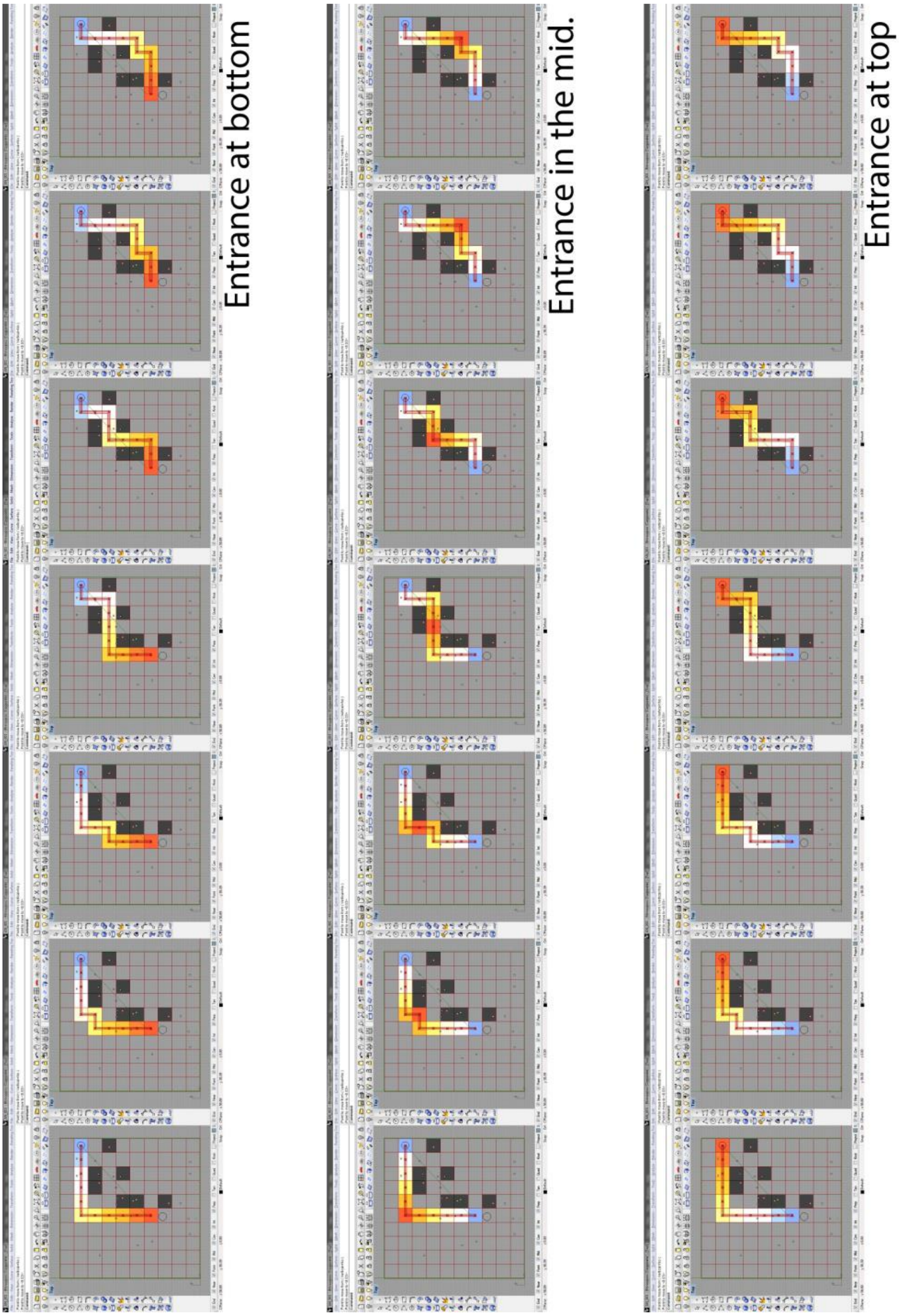


Figure 27: Breadth First Search analysis performed on optional shortest paths generated by ANG (drawn by the author)

As a conclusion this chapter studies the mathematical background of the proposed computational model through formulations and diagrams. The steps studied above related to the proposed model uses several different techniques and methods to perform its generative and analytical features. The important part that needs to be pointed out is that the SP and TSP components are developed in the scope of this thesis. On the other hand the average distance, breadth-first search and the centrality analysis are implementations of the existing algorithms to the proposed multi-layered model.

In order to clarify and position the use of ANG inside the architectural domain, the development process, the data structure and the implementations of the model on specific case studies will be illustrated in the following chapters.

CHAPTER 4

THE USABILITY OF THE MODEL

4.1 Development Process of the ANG

The motivation behind the development of the ANG is based on the difficulties and insufficiencies of the methods and tools that have been used on my graduation project in TU Delft in CompArch studio conducted between TU Deflt and METU. The quality and the content of the ANG have been extended during the studies of the research thesis in METU. The need for a computational analysis tool that can work with the current generative software for specific purposes was clear. The development process of ANG followed many steps listed below:

a. Common characteristics of temporary space designs determined
b. The relationship between temporary spaces and their context questioned
c. Related events that take place in temporary spaces listed
d. General behaviors of temporary space users are determined
e. The conceptual structure of ANG was developed to test the responsiveness of the model to the changes from the environment
f. A flowchart was developed to map the relationships between the inputs form the environment, hybrid analysis tool and generative system
g. Decisions made by the designer and the computer generated decisions are determined
h. Several algorithms are evaluated and the relevant one is determined
i. The general structure of AN was formed based on the conceptual model
j. Some of the components used in Grasshopper software restructured and transferred into code
k. A usability test was conducted to test the usability of the tool
l. In order to assess the validity, reliability and precision of ANG, convergence tests were performed through several case studies.

4.2 Data Structure of ANG;

Firstly, the boundaries of the study are determined as temporary space design. Second, design decisions and common constraints affecting the temporary space designs are explored. These design decision and constraints are taken into consideration to clarify the outline that will form the framework of the conceptual structure of ANG.

- **Site related characteristics;** Site boundaries and topography
- **Event related characteristics;** Event typology, user behaviors and needs, event related functions
- **Infrastructure related constraints;** Infrastructure for temporary space design
- **Features to analyze;** Average distance data, contiguity conditions, betweenness centrality analysis and cellular conditions regarding occupancy,
- **Generative Features**; Combination of all above mentioned properties in a multi layered structure, which is capable of generating possible design decisions such as shortest walk options, shortest path to entrance/exit, shortest path for infrastructure and 'traveling salesman' path analysis.

A concept diagram showing how decisions are made considering the fundamental objectives of the computational model is presented in figure 21. The structure of the proposed computational model is based on the above mentioned, five decisions. The first group is related to characteristics of the site, which are mainly environmental and physical conditions of the site. These are mainly the local characteristics such as physical boundaries and constraints of the site.

Second group contains event related characteristics and decisions. With respect to the limits of the research aimed in this thesis, related events considering temporary space design is listed as; exhibition, extension, commercial, concert, party, shelter and parking. This list can include many more other events considering temporary space requirements. This group also examines the grid typologies in order to suggest the designer possible options for grid mapping. Possible grid selection options for mapping contain two main categories; regular grids and irregular grids. Regular grid represents any Euclidian geometry in a regular pattern such as rectangular grid, hexagonal grid and triangular grid. On the other hand an irregular grid represents simple geometries in an irregular pattern such as; voronoi grid, irregular triangular tessellation. All these grid typologies are dependent on designer decision but related to event type and responsive to environmental parameters.

The third group is based on the necessity of infrastructure of temporary space designs. This group examines the shortest path options in order to provide a set of optimized path solutions to the designer for generating infrastructure for temporary spaces.

The fourth group deals with the computational analysis. This group starts with occupancy analysis which checks the cells of the applied grid for obstacles resulting two new sets of cells; active/unoccupied cells and de-active/occupied cells. Second analysis is the average distance analysis which looks for the average distances of one cell to all other cells. This analysis results in a color diagram in addition to the metric list of distances. Third step is called breadth-first search analysis. The output of this analysis helps the designer to better understand the contiguity conditions of the active cells.

The last group mainly combines all the supplied and analyzed data from different layers of the system. This data can be used manually by the designer or supplied to the generative components of ANG. These generative components are; SP (shortest path), and TSP. Shortest path algorithm is developed as a component for the generative software Grasshopper. This tool basically generates a list of possible shortest paths from a starting point to the destination point. TSP (traveling salesman problem) is the second component developed in the scope of this thesis again as a component of Grasshopper. TSP performs a more advanced version of shortest path search which generates a list of shortest paths that goes through all the desired active cells.

4.3 SP (Shortest Path) Component

SP component is developed as a component for Grasshopper to find the shortest path between two given points on a cellular organization to create multi-cellular temporary design options. The component is developed by Eser Özvataf, Cem Meydan, Anıl Üşümezbaş and Bilge Göktoğan. This computational tool is developed specifically for the use of designers working on temporary design topic. Grasshopper as a generative plug-in program of the software Rhino supports several programming languages. The program development environment has been chosen as C#.

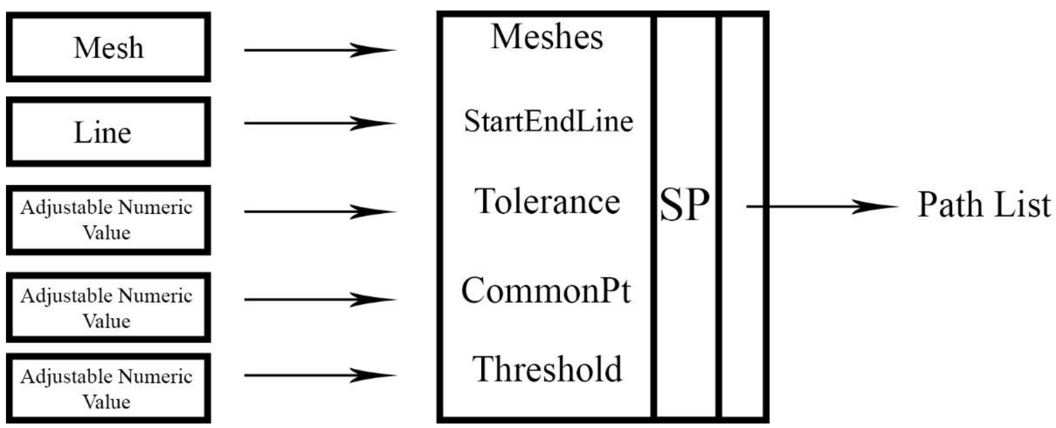


Figure 28: Inputs of the Shortest Path Component (drawn by the author)

4.3.1 Inputs of SP;

Shortest Path (SP) component asks for 4 inputs to be supplied;

1. M (Grid Mesh); Curves of the grid in mesh format
2. L (Line); The line that supplies the starting/source and end/destination points
3. T (Floating Number); Numeric value to define the tolerance of the overlapping points
4. C (Integer); Numeric values of 1 and 2 in order to determine the common points of a grid cell
5. TH (Threshold); Numeric value in order to determine the threshold for the length of common edges

First, the list of curves containing the active grid cells is converted into mesh and supplied to SP component as the M input. Second, the line representing the starting/source and end/destination points of the desired path is set. Third, the numeric value defining the tolerance is set. This number varies according to the project scale. This input T determines the tolerance of overlapping points in case of a mismatch. The input is C which is an integer defining;

- C=1; means there is only one common point between the cells of the grid. This situation may occur in irregular grids and generally in triangular tessellations.
- C=2; means there are two common point between the cells of the grid. This situation occurs in regular grids.

The last input is the threshold which defines the minimum length of a common edge between two polygons in order to be accepted as neighbors. This gives the designer flexibility to choose the desired edge to set the neighborliness condition.

4.3.2. Outputs of SP;

There are two types of results generated by the SP component. First result is the set of paths generated by SP that starts from the source and ends in the destination. The second result is the list of points in the right order to create that path.

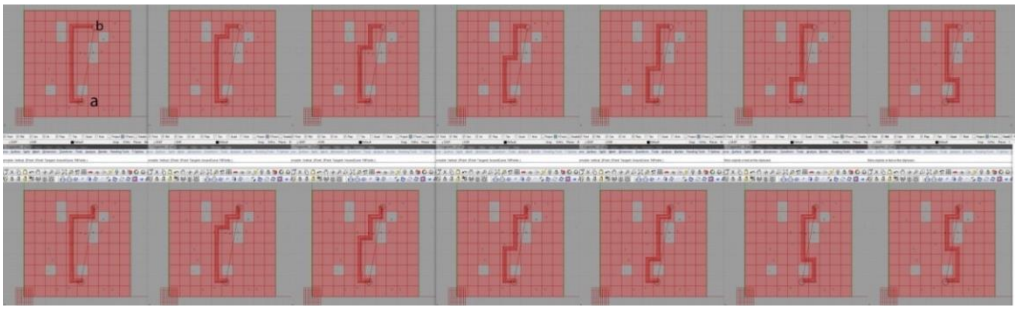


Figure 29: Possible shortest path options generated by SP component (drawn by the author)

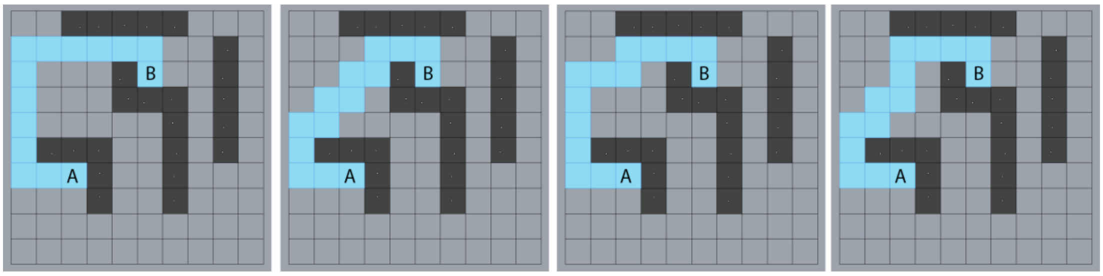


Figure 30:Possible results of SP component in cellular format
Dark cells are the deactive/occupied cells and the blue cells are the generated shortest paths (drawn by the author)

4.4 TSP (Traveling Salesman Problem) Component;

TSP component is developed in order to generate possible minimum detours by visiting every given cell in a multi-cellular organization. TSP is developed as component for the software Grasshopper by Cem Meydan, and Bilge Göktoğan. Grasshopper as a generative plug-in program of the software Rhino supports several programming languages. The program development environment has been chosen as C#.

4.4.1 Inputs of TSP;

As suggested in the conceptual diagram presented in Figure 11, the generative features of the ANG is the last step of the process. This means that many site related decisions have already been made and ready as an input. TSP asks for four inputs to be supplied. The inputs and outputs of TSP is represented in Figure.36

1. S (Starting Points); Starting points of the supplied list of curves
2. E (End Points); End points of the supplied list of curves
3. T (Integer); Numeric values from 1 to 4 in order to decide the path typology
4. R (Integer); Seed for other possible path generations

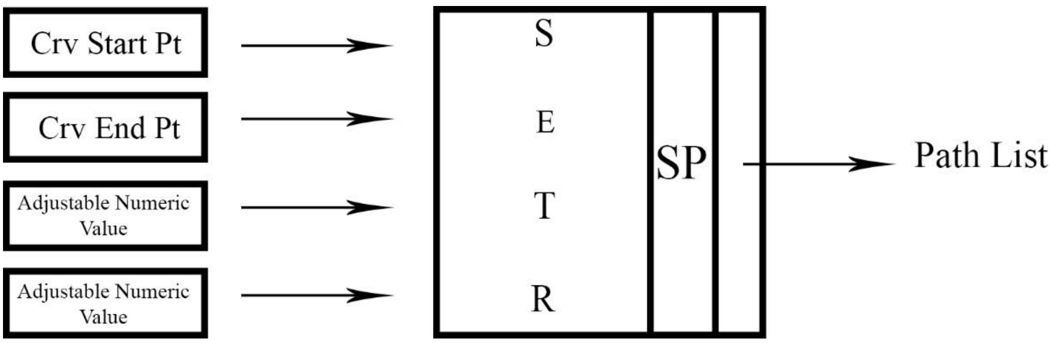


Figure 31: Inputs of the TSP component (drawn by the author)

The curve list which supplies start and end points is a collection of the active grid cells in mesh format. First, these collections of curves are decomposed into vertices in order to be supplied to TSP component. Second, the input T is determined which technically is an integer between 1 to 4.

- T=1; checks for contiguity and assumes distance as 1 and generates a linear path
- T=2; checks for contiguity and assumes distance as 1 and generates a cycle
- T=3; works with Euclidean distance and generates a linear path
- T=4; works with Euclidean distance and generates a cycle

The last input required is R which is the seed source of the TSP. R is an integer like T but its function is quite simple. R is the seed that makes it possible for the user to represent other possible path options starting from the minimum length path.

4.4.2. Outputs of TSP;

There are two types of results comes out from TSP. First result is the path generated by TSP that visits every given cell with a minimum detour. The second result is the list of points in the right order to make that detour.

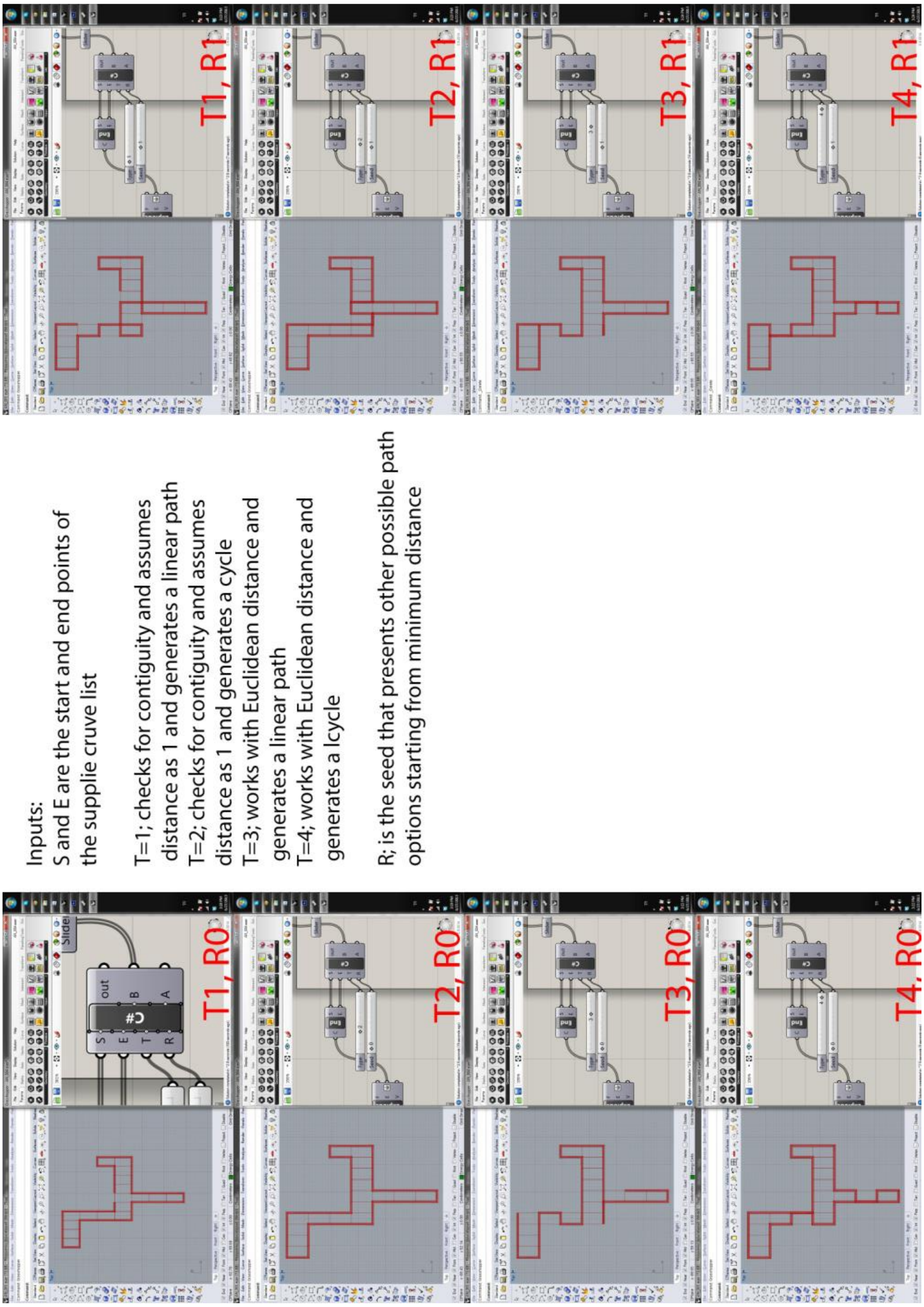


Figure 32: Outputs of the TSP component (drawn by the author)

Figure 32 presents the results of TSP according to four different types of calculations for two distinct generations. These results looks more or less the same visually but the possible ways of calculation that TSP offers creates a promising set of approaches for designers to choose. Depending on designers approach for calculating a shortest detour for a linear or a branched multi-cellular organization it is possible to calculate a linear path or a cycle by both using weighted (using Euclidean distance) and unweighted (assuming all distances are equals to 1) graphs.

The components composing the multi-layered structure of ANG that are studied in this chapter are applied on real cases in the next chapter. The following chapter illustrates and discusses the studied features of ANG over several case studies and presents the results.

CHAPTER 5

CASE STUDIES

The model proposed in this thesis is aimed to be a generative model called 'Adaptable Network Generator'. The primary motivation of developing such a model is the lack of analytical tools to support the so called generative system that is developed in the scope of the graduation project in CompArch design studio in TU Delft. In this chapter the results of the ANG is discussed through several case studies.

The first case that is studied is the project that I have designed in Lange Voorhout in Den Haag in Netherlands in CompArch design studio conducted between METU and TU Delft. The project was designed before the development of ANG so the following study that is discussed is aimed to question the validity of the final design by regenerating the base components of the same project by using the same constraints and parameters.

Second case is a museum design in Ankara called Erimtan Archeology and Arts Museum. The first case is an outdoor application of ANG. On the other hand this case is selected in order to test the results of ANG in an indoor space containing a temporal space organization.

5.1 Pavilion Design Project in Den Haag;

In this project the students were asked to design a pavilion that can host multiple events in multiple locations with respect to the constraints and dynamic characteristics of the given site. The project site was in Lange Voorhout in Denhaag in Netherlands. Several locations in the same site were chosen for hosting different events. In this chapter two of those designs are selected and studied namely; the 'market place' design and the 'extension' design.

Figure 33: Redefinition of the urban grid for Lange Voorhout. (drawn by the author)

In this case study the existing results of the project are compared with results generated with ANG by following the steps below;

- Cellular Mapping; Generating the grid for infrastructure and ultrastructure
- Comparison of the multi cellular pavilion organization for market place design that is done manually with the result generated by the shortest-path component of ANG
- Analytical results generated by the ANG that does not exist in the project that is designed manually in TU Delft
    - Average Distance Analysis
    - Breadth-First Search Analysis
    - Betweenness Centrality Analysis
    - Average Distance Analysis for the Shortest Infrastructure Path
    - Paths generated by TSP

5.1.1 Cellular Mapping;

Cellular mapping as explained in Chapter 3 is the first step for ANG. In the pavilion design study the cellular mapping is performed for both 'market place' and 'extension' designs. The reason behind a common mapping is that they both designed for the same site so the same grid works for both cases.

For this project an irregular triangulation tessellation, called 'Triangular Duo' which is discovered by L. Danzer and C. Goodman-Strauss is chosen. This triangulation is mapped inside the boundaries of the selected site for pavilion design. Necessary adjustments are performed by the designer and the irregular grid for infrastructure is generated (Figure 34).

The reason behind selecting such a grid comes from the motivation for developing a multi-cellular pavilion design that is unique for every location/event and at the same time modular with respect to fabrication purposes. In this perspective Triangular Duo is a perfect choice because it consists of 4 different size/type of triangles but as a grid it provides countless possibilities for multi-cellular organization. That makes it a very promising grid choice as an ultrastructure schema for generating multi-cellular pavilion designs.

The grid that is generated for the infrastructure is a voronoi grid that is using the vertices of the ultrastructure as a reference. That is why the super-position of both grids works in harmony (Figure 34). The voronio grid provides pressured air for the inflatable pavilions and contains fiber optic cables for data and light transfer.
Figure 35 shows the night time of the market place design where the infrastructure containing fiber optic cables are illuminating and transferring light to the pavilions through the inflatable cavities that are also storing fiber optic cables.

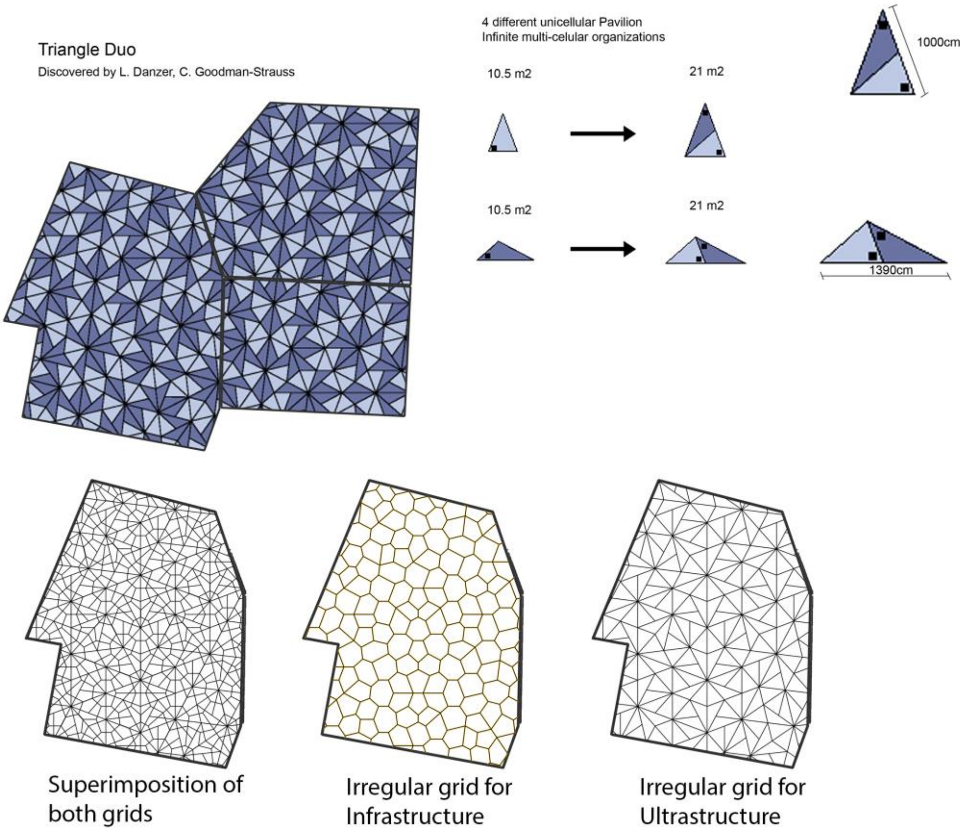


Figure 34: Cellular Mapping for the pavilion design project (drawn by the author)

Figure 35: An image showing the night time of the Market Place with illuminating infrastructure. (drawn by the author)

5.1.2 Comparison of the Original Results to the Results of ANG for the Market Place Design;

Here in this study the cellular organization of the market place design of the original project designed in TU Delft (Left image in Figure 37) is compared with the results of the ANG (Right image in Figure 37). The aim behind this specific organization is to regenerate a market place in Lange Voorhout with the proposed system. This system aims to generate multi-cellular pavilion design solutions using the grid that is mapped on the site.

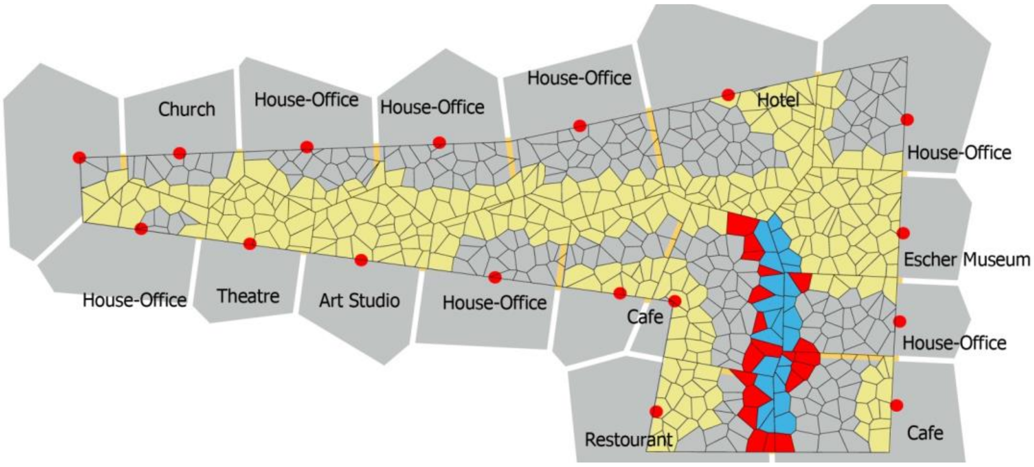


Figure 36: Site plan showing the possible locations for market place design. (yellow: all possible locations for a pavilion design, red: possible locations for market place, blue: manually selected locations for market place design)(drawn by the author)

The image on the left in Figure 37 is the result of this process and it is generated by selecting the desired neighboring cells manually. The fundamental condition of this manual selection is that all cells should be adjacent to each other in order to create a working multi-cellular organization.

The image on the right in Figure 37 represents the results of the ANG. In this image the obstacle detection and shortest-path generation steps of ANG are represented. As it can clearly be observed in Figure 37 ANG generated the same result with the original project in which the cellular organization was decided manually. The existing trees on the site are defined as obstacles so in the obstacle detection stage of the model those cells are marked as occupied and removed from the active cells list (the black cells in Figure 37). So in the shortest-path generation stage only the unoccupied cells are taken in to calculation. Two separate multi-cellular organizations where all the cells are adjacent to each other are generated following the shortest-paths calculated from A to B.

One of the crucial aspects of ANG is to generate all the possible shortest-path options but in this case the grid is an irregular grid so there is just one shortest-path result which is identical to the manually designed one.

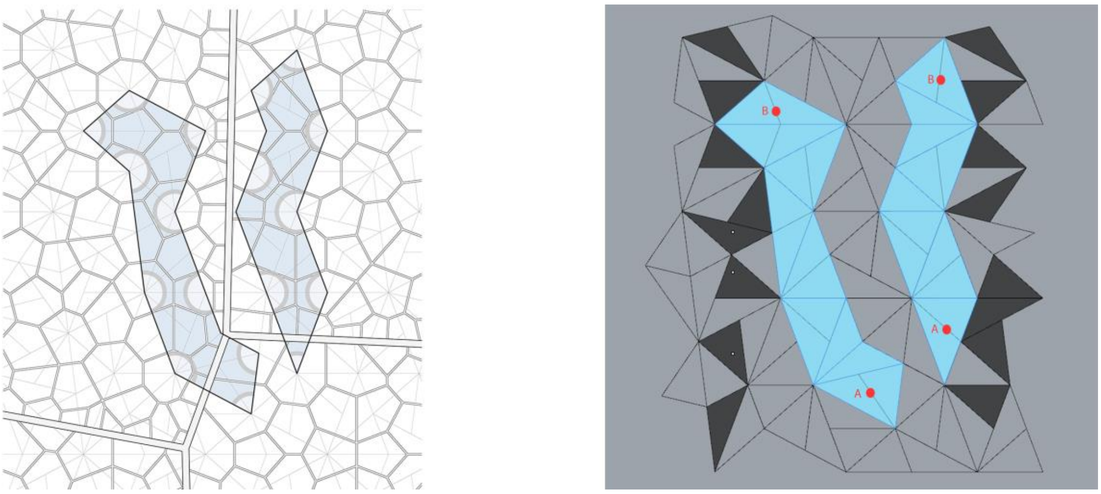


Figure 37: The comparison of the original project (Left), with the result of ANG (right) (drawn by the author)

ANG is aimed to generate possible shortest-path options for multi-cellular temporary space organizations for conceptual design phase. The final result of the pavilion design for market place area in Lange Voorhout with respect to the generated multi-cellular organization is represented in Figure 38 and Figure 41.

Figure 38: The final image showing the multi-cellular pavilion design for market place area (drawn by the author)

In Figure 41 the same site which was selected for market place design is used. The obstacles (trees in that case) are in the same location as Figure 39 but instead of applying an irregular triangular tessellation for cellular mapping this time a regular rectangular grid is chosen. Figure 41 represents 6 of the possible 35 multi-cellular organizations generated by SP component following the shortest paths from point A to point B with respect to the shortest infrastructure path concern.

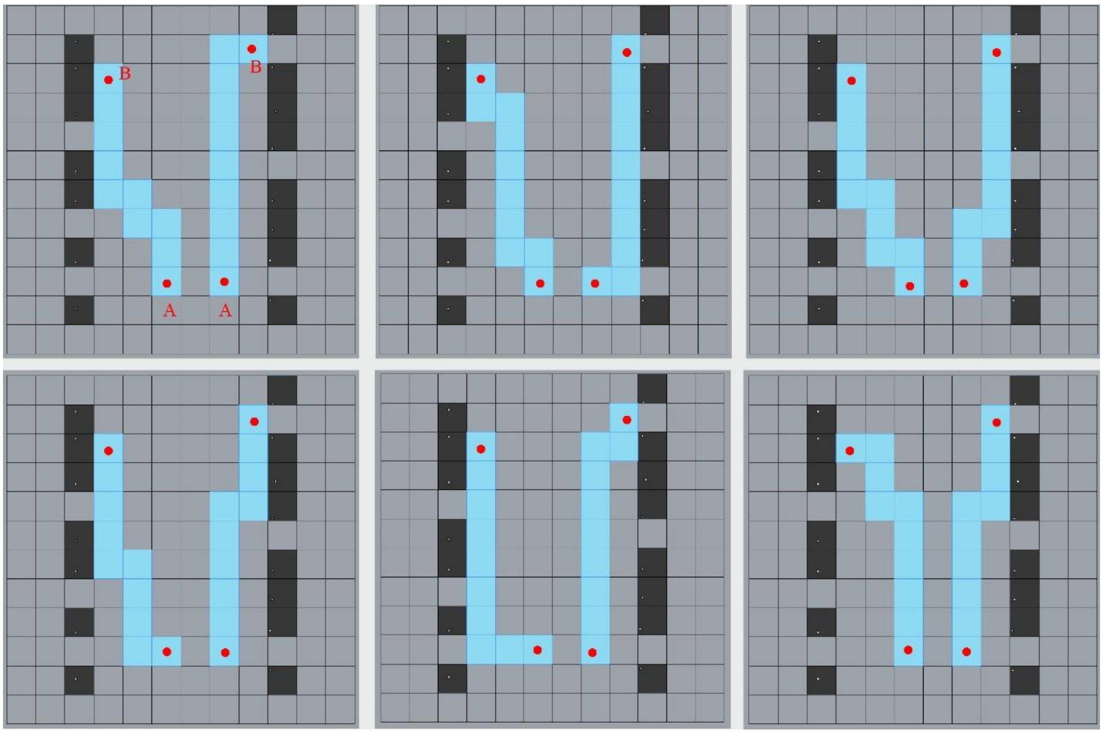


Figure 39; possible multi-cellular organizations generated by ANG (drawn by the author)

In order to have a better understanding about the potentials of the ANG and its generative features the same site is studied in
Figure 40.
Figure 40 shows 6 different multi-cellular organizations on a hexagonal grid. These organizations also follows the shortest paths generated by SP component. The results shows that not only linear paths but also branched organizations can possibly be generated by using ANG.

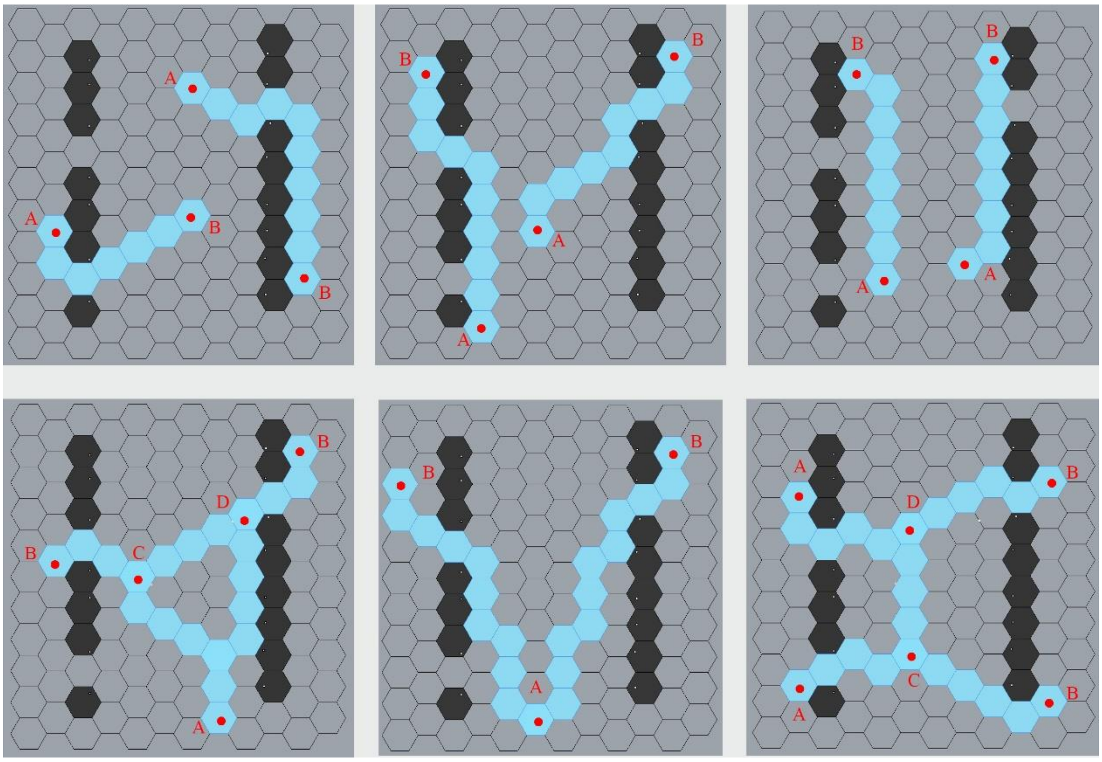


Figure 40 ; Multi-cellular organizations generated on a hexagonal grid by ANG
(drawn by the author)

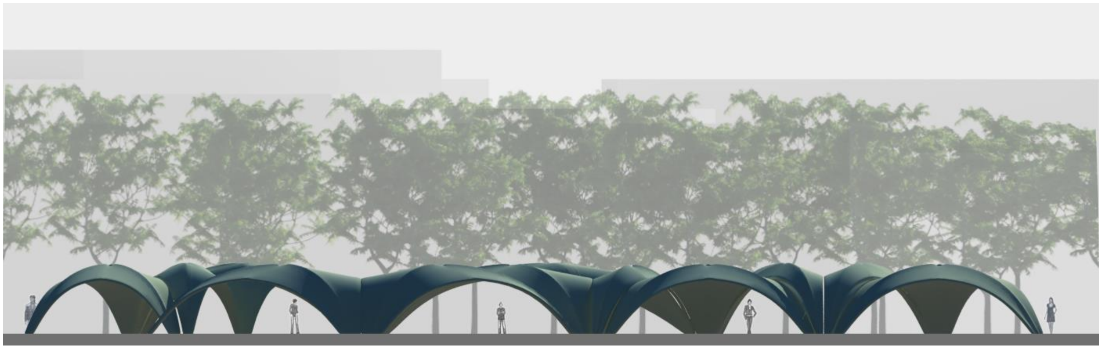


Figure 41: Elevation showing the multi-cellular pavilion design for market place area (drawn by the author)

5.1.3. Analytical Results Generated by ANG for 'Market Place' Design;

The analytical result studied here does not exist in the original project so the discussions will not be performed in comparison to the original project. The studied analytical result are;

- Average Distance Analysis
- Breadth-First Search Analysis
- Betweenness Centrality Analysis
- Average Distance Analysis for the Shortest Infrastructure Path
- Paths generated by TSP

5.1.4. Average Distance and Breadth-First Search Analysis;

Average distance and breadth-first search analysis that are studied in detail in Chapter 3 are used in this case study to collect further information about the relationships of the components constructing the multi-cellular organization of the market place design. The analytical results of the market place design is compared with the average distance analysis of a generic branched organization for the same site in Den Haag. Designers can benefit these analytical readings of the generated organization in many different aspects depending to their approach. For example Figure 42 represents the result of average distance analysis from an entrance in the center to all other cells. Left image represents the result belonging to the original market place design (generated by ANG) and right image represents the result which is conducted on a generic branched organization for the same site with the same constraints.

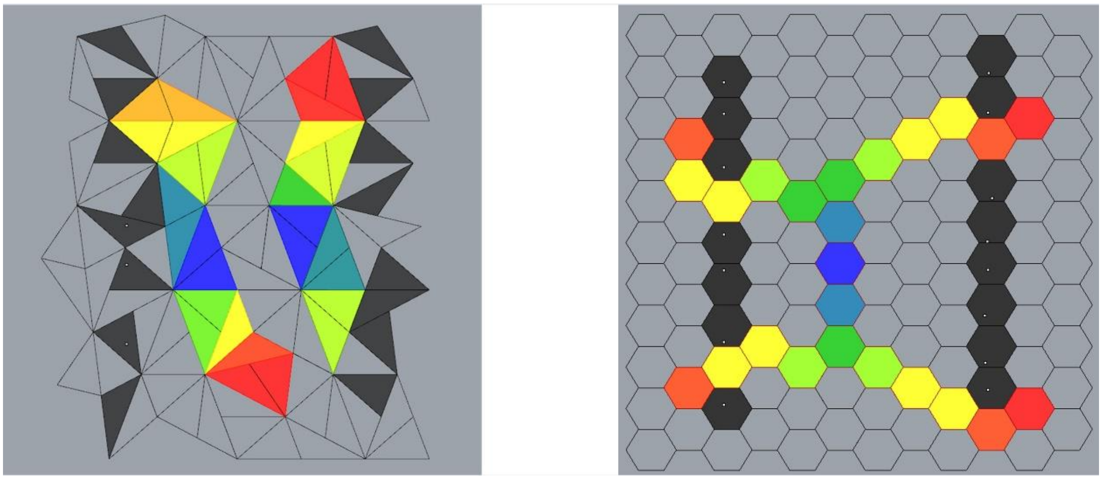


Figure 42: Average distance analysis from an entrance in the center to all other cells (Left) Average distance analysis from an entrance in the center to all other cells on a

hexegonal grid order with branched organization (Right) Hue represents the average distance (blue=min and red =max) (drawn by the author)

Figure 43 shows the results of the breadth-first search analysis from every cell to all other cells. Left image show the analytical date belongs to the generated organiztion with respect to the market place design and right image shows the same analysis on a generic organization on the same site. This analysis works as an unweighted graph in other words it does not use euclidean distance and assumes distance between the nodes as 1. Thats why the reading of the neighbourhoodness of the components can be done.

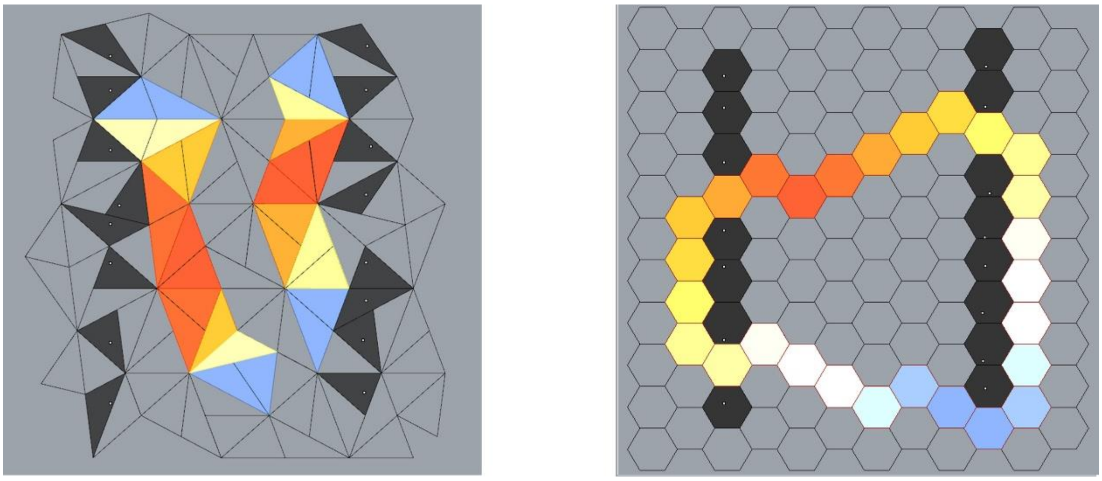


Figure 43: Breadth-first search analysis from every cell to all other cells for market place design (Left) Breadth-first search analysis from every cell to all other cells for a generic organization on a hexegonal grid (Right) Hue represents the adjacency (light blue=min and orange =max) (drawn by the author)

5.1.5 Betweenness Centrality;

Betweenness centrality analysis helps the designer to quantifiy the number of times a cell acts as a bridge in a multi-cellular organization. In the case of market place design

Figure 44 represents the centrality conditions of the components of the generated organization.

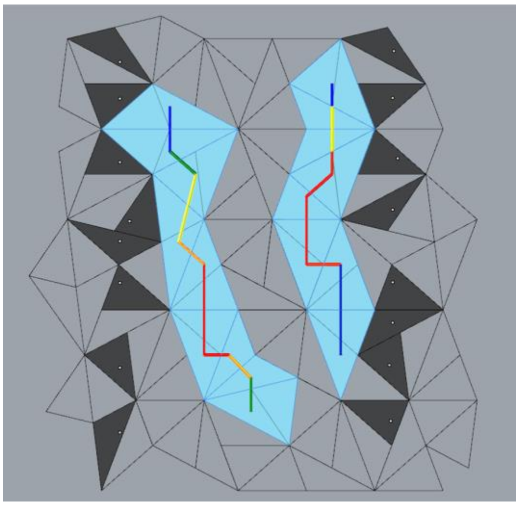
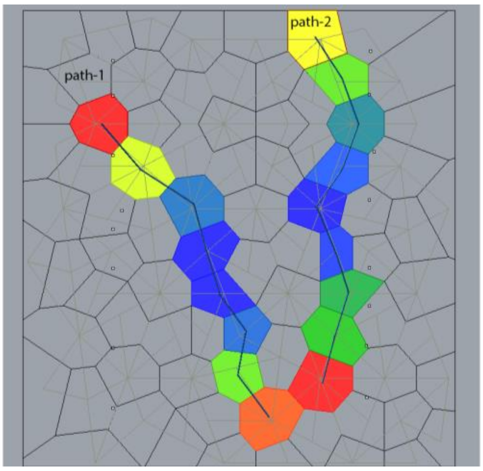


Figure 44: Betweenness centrality analysis of Market Place organization (blue=min and red =max) (drawn by the author)

5.4.3. Avarage Distance Analysis for the Shortest Infrastructure Path;

Many temporary space designs are dependent on an infrastructure to function properly. The pavilions in this project are designed as inflatable structures with fiber optic cabels integrated to their inflatable cavities. The infrasture design works as a network that is distributed homogeniusly to the whole site. Figure 45 shows the shortest infrastructure path to support the pavilion designs on top of it.



Metric Avarage Distance List

Path-1	Path-2
250	699
188	484
147	910
132	605
132	515
178	646
146	528
232	571
	550

Figure 45: Average distance analysis of the shortest-path for infrastructure (Left) *Metric average distance list of the paths generated (Right) (drawn by the author)*

5.1.6. Paths Generated by TSP;

TSP component works with the edges of a given set of closed polygons and generates possible shortest detours visiting all the given cells in a multi-cellular organization. In the case of market place design all possible outcomes of the component are tested and visualized. In Figure 46 two possible shortest-paths visiting all the given cells are represented. In these two alternative paths, the distances between the cells are assumed as 1 (unweighted graph is used) and linear paths are generated. In more case specific terms this can also be explained as; a visitor can visit all the market area by following these shortest-paths.

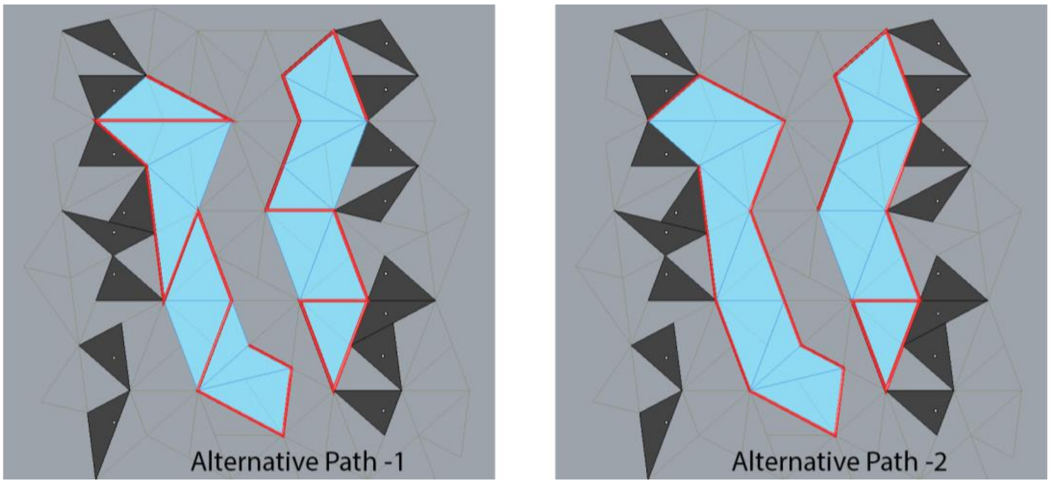


Figure 46: Two alternative linear paths generated by TSP component assuming the distances between cells are 1. (drawn by the author)

On the other hand Figure 47 shows again the shortest detour for market place organization but using the Euclidean distances to calculate the shortest-path.

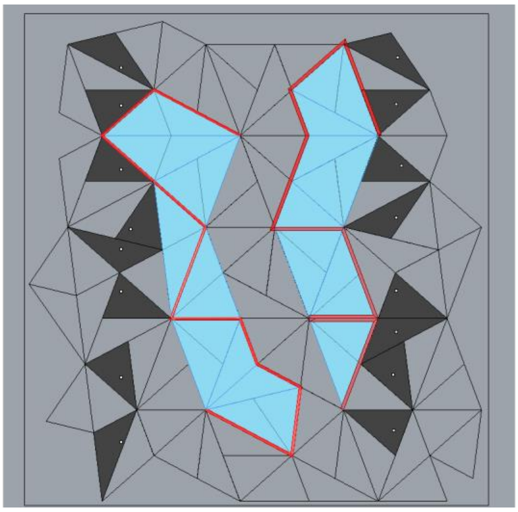


Figure 47: Linear path generated by TSP component using Euclidean distances (drawn by the author)

Figure 48 shows the results of a full cycle generated by TSP component for visiting the original market place organization (left) and the generic branched organization on a hexegonal grid. It should be noted that the results of TSP when calculating cycles (in linearly formed multi-cellular organizations and also in branched ones) generally follow the outer boundaries of the organization.

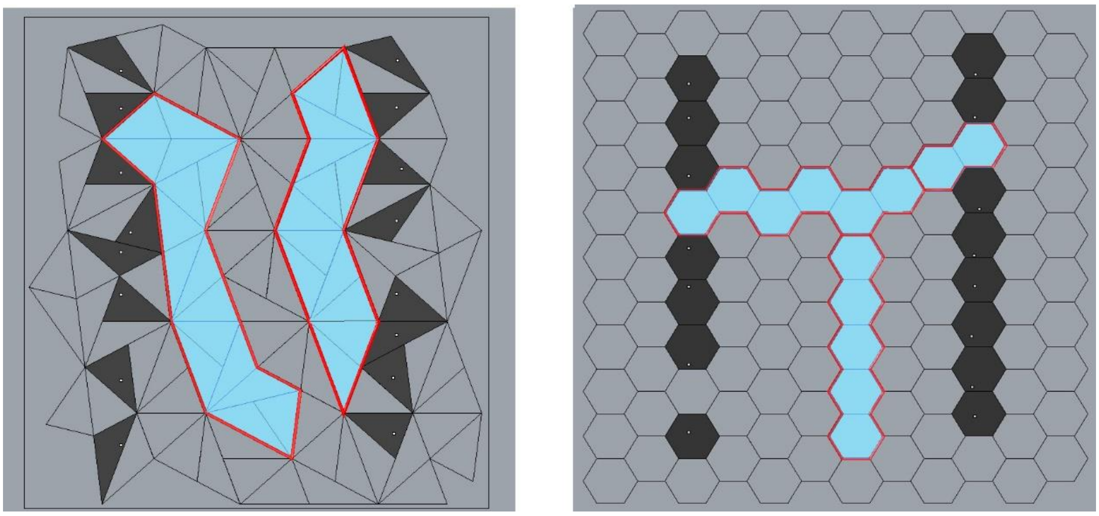


Figure 48: Cycle generated by TSP component assuming all the distances equal to 1 (Left) Cycle generated by TSP component using euclidean distances (Right) (drawn by the author)

5.2. Comparison of the Original Results to the Results of ANG for the 'Extension'

Design;

This case study belongs to the same project conducted in TU Delft CompArch design studio. It is called the 'extension' pavilion. The aim of the 'extension' is to design a multi-cellular pavilion organization in case of an existing building in Lange Voorhout demands extra outdoor space to facilitate an event. In this specific case of the extension pavilion the purpose is to create a cocktail area in front of the Pulchri Art Studio in Lange Voorhout.

Figure 50 shows the comparison between the original design and the result of ANG. According to the demands of the art studio necessary size of the cocktail area and the location of the pavilions are decided. The left image in Figure 50 shows the manual selection of the suitable neighboring grid cells.. The right image shows the result ANG calculated by the SP component. In the case of the extension design ANG again validates the correctness of the original project.

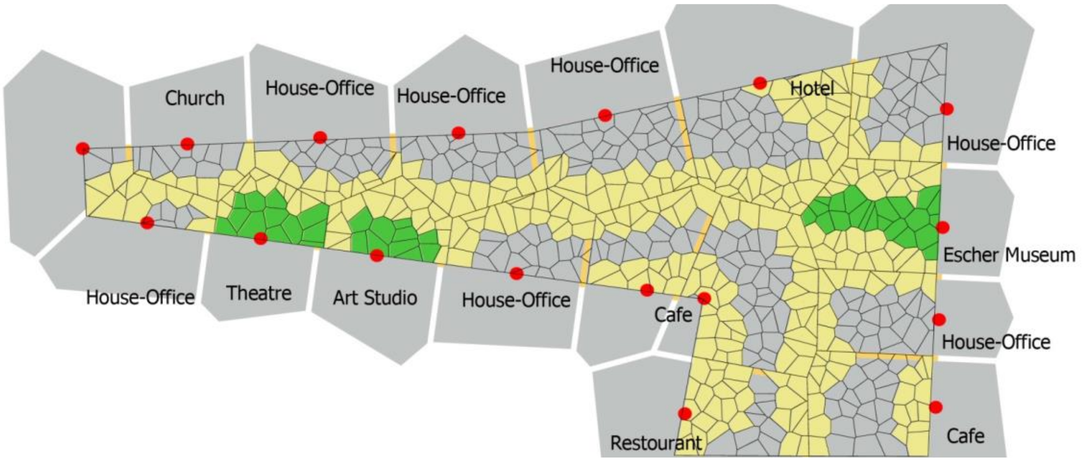


Figure 49: Site plan showing the possible locations for extension design. (yellow: all possible locations for a pavilion design, green: possible locations for extension design)(drawn by the author)

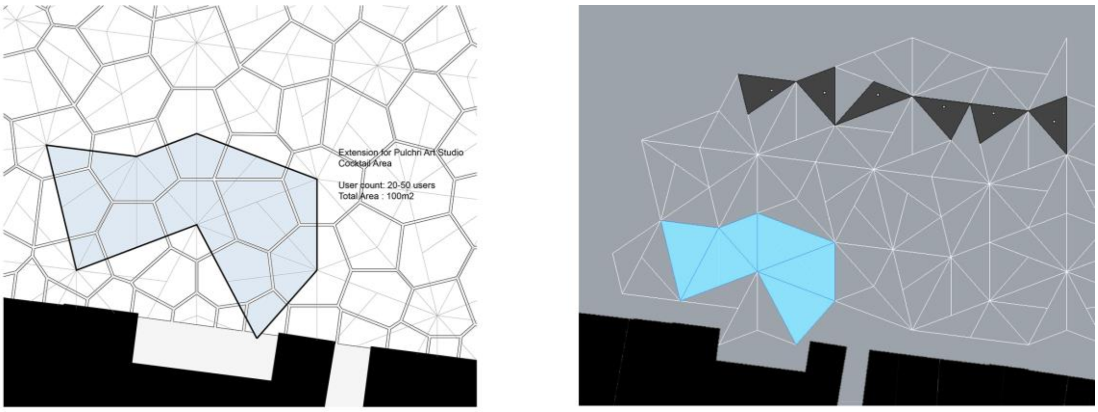


Figure 50: Showing the comparison of the original project (Left), with the result of ANG (right) (drawn by the author)

Figure 51 show the 3D visualization of the generated pavilion design for Pulchri Art Studio.

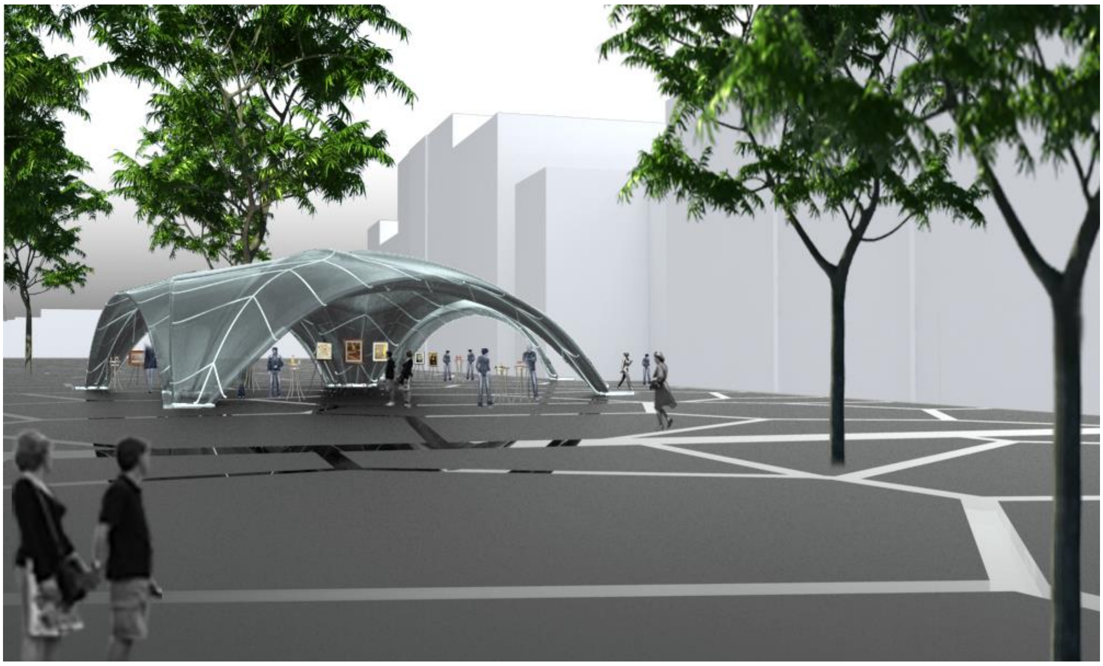


Figure 51: Day time view of the extension design for Pulchri Art Studio. (drawn by the author)

5.2.1. Analytical Results Generated by ANG for Extension Design;

Extension design is a smaller scale multi-cellular organization consists of six cells. The studied analytical results are;

- Average Distance Analysis
- Breadth-First Search Analysis
- Betweenness Centrality Analysis
- Path generated by TSP

The results listed above are discussed through singular outputs with respect to the scale of the case. The discussion containing all the possible outcomes of ANG is studied in the case of market place design in the previous case study.

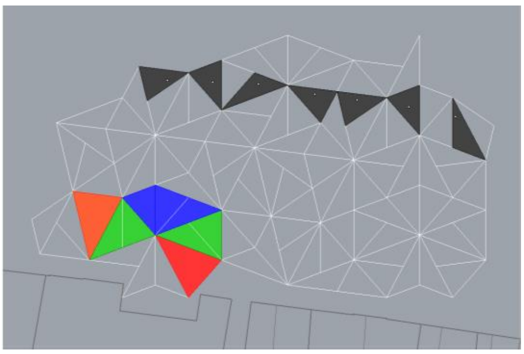


Figure 52: Avarage Distance Analysis from every cell to all other cells (drawn by the author)

In Figure 52 the average distance analysis (right) is represented. This data is useful for the distribution of the program elements for the pavilions that are generated to facilitate a cocktail area.

On the other hand Figure 53 represents the betweenness centrality and TSP analysis. The data related to the centrality condition can be useful for the positioning of the art pieces that are planned to be exhibited during the cocktail. The result showing the shortest detour generated by TSP component is represented in Figure 53 on the right. It shows the shortest circulation path for the visitors to visit every pavilion.

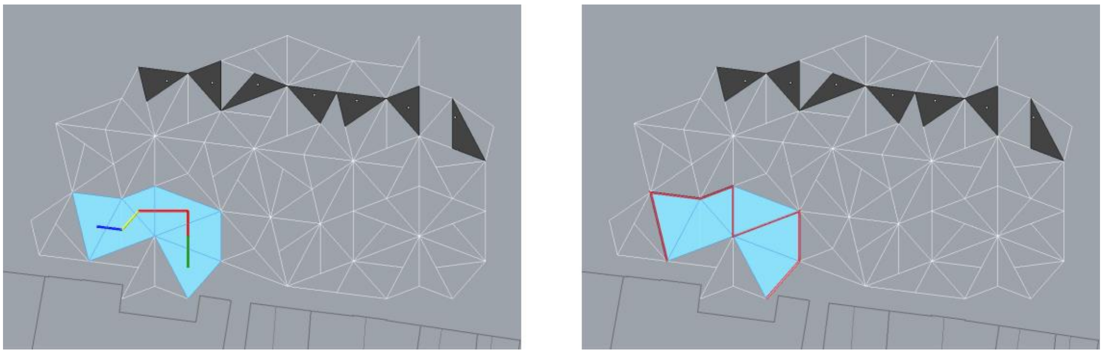


Figure 53: Betweenness centrality analysis (Left) and shortest detour generated by TSP (right) (drawn by the author)

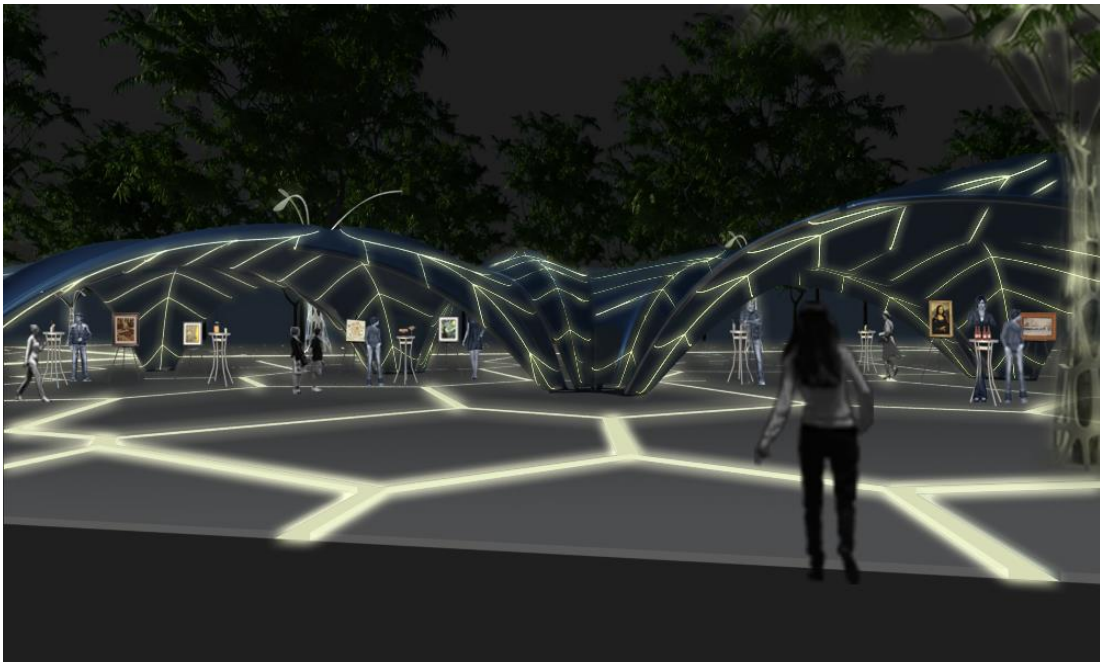


Figure 54: Night time view of the extension design for Pulchri Art Studio from the entrance of the studio. (drawn by the author)

5.3. Analytical Results Generated by ANG for Erimtan Archeology and Arts Museum;

Erimtan Archeology and Arts Museum is analyzed using ANG in order to test the results of the computational model on an indoor space with more concrete boundaries and obstacles. Unlike the previously studied two cases this case contains the analysis of the whole site in addition to the analysis of the shortest-paths generated.

First a regular rectangular grid is mapped on the floor plan of the museum. The separator walls, staircase and the gallery cavities on the slab are marked as obstacles. The black cells represent the occupied locations which are removed from the active cells list. In other words the black cell are the locations where no exhibition can take place. Figure 55 shows the average distance analysis that is performed on the whole floor plan.

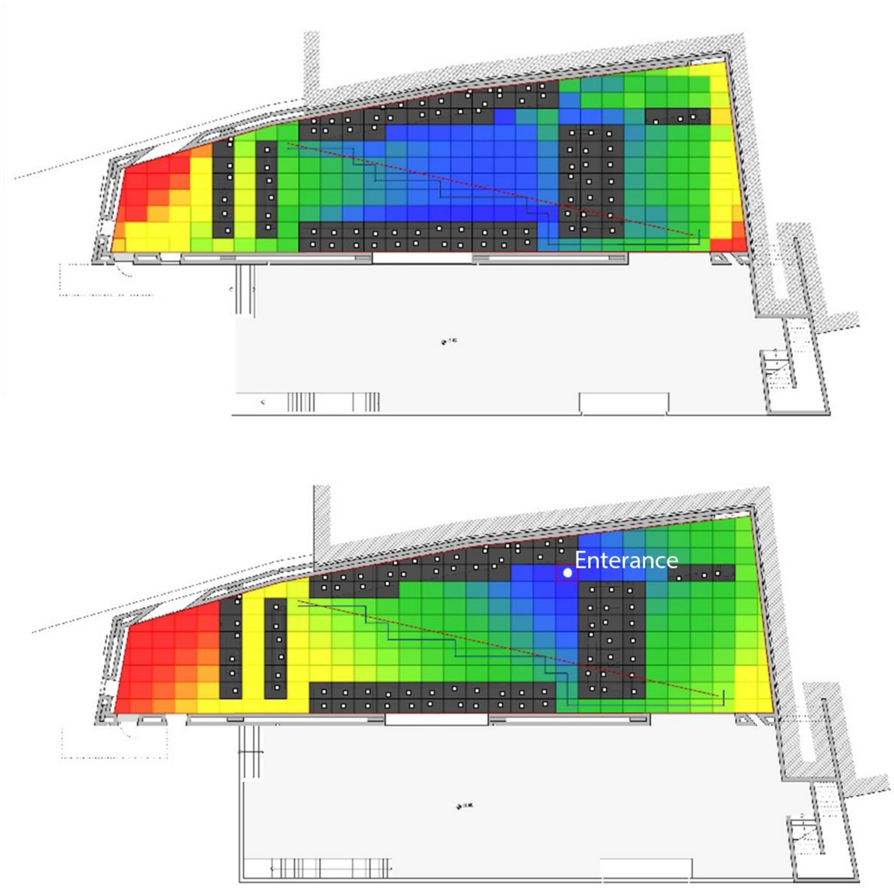


Figure 55: Average distance analysis from every cell to all other cells, performed on the whole floor plan (Top). Average distance analysis from entrance to all other cells, performed on the whole floor plan (Bottom) (drawn by the author)

Left image shows the result from every cell to all other cells which gives the designer an overall idea about the use of whole area and the distances of specific locations on a cellular schema. The image in the right shows the avarage distance analysis from entrance to all other cells. This analytical reading gives a designer an insight about the distances of specific locations from the starting point of a tour in the exhibition area so that the exhibition can be designed accourdingly.

Figure 56: Betweenness centrality analysis performed on the whole floor plan (drawn by the author)

Figure 56 represents the betweenness centrality analysis performed on the whole floor plan of Erimtan Archeology and Arts Museums exhibition area where the centrality condition is quantified over the number of times a node acts as a bridge. In comparison to the avarage distance analysis visualized in Figure 55 (left) this data points out the difference between the metric centrality and the centrality in a circulation. This analytical reading can be interpreted by the designer in order to decide the distribution of the art pieces according to the centrality conditions.

Figure 57: Average distance analysis from every cell to all other cells, performed on a generated shortest-path (Left). Average distance analysis from every cell to all other cells, performed on an other shortest-path (Right). (drawn by the author)

Figure 57 represents two different avarage distance analysis from every cell to all other cell on the generated shortest-path. Such a path generation and the analytical readings on that path can be useful to understand relationships of individual locations between two specific points in the museum.

CHAPTER 6

DISCUSSIONS AND CONCLUSION

This chapter provides a summary of the arguments that are constructed in this thesis. It presents discussions about the computational model developed in this study namely; Adaptable Network Generator (ANG). This chapter also discusses the potentials and the limitations of the proposed model with respect to the temporary space creation which suggests further studies and draws some conclusions and presents an overview of the study. It evaluates the methodology of the thesis in addition to the contributions and limitations of the proposed model and suggests further studies.

6.1. Methodology;

The necessity to investigate the topics covered in this thesis started with the problems that the author faced with in his graduation project. Besides developing solutions to those problems, this thesis investigates adaptive architectural design solutions for complex spatial organizations such as temporary spaces by using a generative approach. The result is a computational model called Adaptable Network Generator (ANG) that generates circulation paths, cellular organizations and connections as networks for a given site and a given context.

Temporary space creation requires parallel characteristics to evolve in between the desired physical space, environmental stimuli and the shifting programmatic requirements. This thesis proposes a multi-layered computational model in order to analyze and generate organizational design solutions with respect to temporary space design. These layers are; the generative layer and analysis layer.

In order to develop such a tool, the author inquired into three main topics which are generative systems, temporary space organizations and spatial analysis. The subject matters related to generative systems are discussed through sub topics of computational thinking, algorithmic thinking, generative design and emergence with respect to the case conducted in CompArch design studio in TU Delft. Some components of ANG are developed in a parametric design environment that is why parametric design topic is also studied.

As the name of the proposed model dictates, ANG (Adaptable Network Generator) generates connections as networks. In order to have a better understanding about the network typologies and characteristics, the definition of network and the idea of network in different fields are explored and questioned.

All the topics in this thesis are studied in order to develop a computational approach for temporary space design and the dynamic characteristics of such cases. Therefore the temporary space organizations are investigated. Urban dynamics of a site are explored from a computational point of view. The author examined the reciprocal relationships of urban patterns as components of a systems and a system as a whole and structured the bidirectional workflow of his approach accordingly.

The multi-layered structure of ANG contains analytical steps where several spatial analysis techniques are used. The computational spatial analysis methods used in architectural and urban design are inquired. The theoretical and practical features of Space Syntax, Cellular Automata, Graph-based Analysis, Spatial Mapping and Artificial Neural Networks are explored. The knowledge learned from these researches are used to create the framework of the analytical layers of ANG.

Author set the objectives of his model and represented the formulation of the components used in the model in a sequence that the computational process follows. The steps of cellular mapping, obstacle detection, shortest-path generation, TSP (traveling salesman problem) analysis, average distance analysis, breadth-first search analysis, betweenness centrality analysis are explained. The formulation of the algorithms and the contributions done during the implementation of these algorithms to the developed components namely; SP (shortest path) and TSP components are presented.

In order to explain the usability of the proposed model, author presented the development process and the data structure of ANG. The generative components (SP and TSP) with their inputs and outputs are represented and the applications of these components in practice are explained. Generative systems offer variable outcomes for different cases. The generative potentials of ANG are tested over different generic grid typologies and author represented the variable outcomes of the model that are generated according to the adjustments offered by the developed components.

The applicability of the computational model developed is discussed over several case studies one of which is author's graduation project designed in CompArch design studio conducted between TU Delft and METU. As a conclusion, in this chapter author summarized his achievements so far.

6.2. Applicability of the ANG;

In order to have deeper understanding on how temporary spaces works and what kind of an approach should be followed, the temporary space organizations are studied and author's graduation project is reevaluated. The theoretical research and the knowledge that author acquired from his graduation helped him to structure his computational approach. As a designer author adopted an approach which dictates that a system should be adaptive to environmental changes and should take advantage of these shifts by responding and adapting to the local and global stimuli. This adaptive approach reciprocally generates changes in the local environment. With this perspective a computational model that is capable of performing preliminary analysis of the selected site, generating multi-cellular organizations following the shortest path options and reanalyzing the generated results is proposed. This approach can be used by designers to create reciprocal relationships with their design and their design environment by grounding their work on analytical data in addition to their intuitive decisions.

In the scope of this thesis author designed two new generative components to support designers who are willing to use a computational approach when designing temporal space organizations. The designs of these components are translated into programing language by professional programmers and implemented into the proposed computational model (ANG) by me. After the components are programmed and practiced in different case studies the applicability and adaptability of the model in the design process of a temporary space is assessed.

6.3. Potentials and Limitations;

The aim of this thesis is not to propose a definition of a new design process for temporary space organizations or to dictate the use of generative design systems is indispensable. The author illustrates a computational approach for a specific case and proposes a computational model that offers generative features with analytical tools to support that approach and its advantages.

Author does not suggest that design processes such as temporary space design cannot be handled with non-computational approaches but he implies that process with so many changing design variables can be better controlled and designed with analytical results and generative systems that work with such data. Instead of doing these analysis and generating design organizations manually author proposes a computational model that can combine these steps and generates possible design organizations.

The computational model proposed in this thesis has a number of potentials such as the multi-layered structure that connects the analytical data with generative components and the generated data with the analytical tools. This model is designed as a parametric model that gives the designer the ability to do adjustments by changing parameters which creates a responsive relationship between the designer and the model. The generative components (SP and TSP) can generate multiple possibilities of multi-cellular organizations and circulations which gives the designer the flexibility to evaluate possibilities from a set of variable results before making further design decisions.

On the other hand the proposed model is developed and improved according to the feedbacks that author got from his applications on the generic and specific cases he conducted. Therefore in such an old topic as temporary space design with a lot of different approaches and designs the contributions of ANG to the field is limited with the aims of this thesis. Another limitation of the model is that it works with cellular organizations to perform shortest-path analysis which means that ANG always deals with bigger point clouds in quantity when compared to a standard shortest-path algorithms. This creates a limit of maximum cells to work with which naturally limits the size of the projects that ANG can be applied or obliges the designer to separate big projects into smaller pieces and work on them individually.

6.4. Contributions of This Study;

The primary contribution of this thesis is the proposed computational model. More specifically the SP and TSP components developed with respect to the temporal space organizations. 'Shortest-path' and 'traveler salesman problem' algorithms have a very large span of different applications in many fields. In this thesis these algorithms are studied and implemented into the context of temporal space design. In the scope of this thesis these algorithms are transformed into SP and TSP components both of which works with closed polygons (grid cells) instead of working with non-spatial point clouds. In addition to that these components checks for adjacency condition and generate organizations accordingly, which makes a small but important architectural contribution to the generation process which does not exist in standard examples.

In addition the multi-layered structure of the proposed model offers a potential set of resultant data to be read where the generated organizations can be evaluated by the analytical data over a set of possible outputs. This way the designer can benefit from the variety of end results.

6.5. Suggestions for Further Studies;

I would like to conclude this thesis by suggesting some research questions and further studies regarding the limitations of the model proposed in this study.

The computational model proposed in this study works inside the site boundaries on a mapped grid, this thesis suggests to further investigate the potentials and possibilities of such an approach on a 3D topology instead of a 2D mapped grid.

For further studies, I suggest working with 3D cellular organizations and volumetric analysis which will potentially open new fields for application in addition to temporary space design.

REFERENCES

Gross, J., & Tucker, T. (1987). *Topological graph theory*. Wiley Interscience.

Louis, S. J. (2010). Simulated Evolution and Learning. *Simulated Evolution and Learning: 8th International Conference.* Kanpur, India: Proceedings on Lecture Notes in Computer ... Computer Science and General Issues.

Akyildiz, I., & Rudin, H. (2014). Computer Networks. *The International Journal of Computer and Telecommunications Networking*.

Alexander, C. (2011). Systems Generating Systems. In A. Menges, & S. Ahlquist, *Computational Design Thinking* (pp. 60-65). Chichester: John Wiley & Sons,.

Bafna, S. (2003). SPACE SYNTAX A brief introduction to its logic and analytical techniques. *Environment and Behavior*, 17-29.

Barrios, C. (2005). Transformations on Parametric Models. *Computer Aided Architectural Design Futures 2005* (pp. 393-400). Springer Netherlands: Computer Aided Architectural Design Futures 2005.

Barthélemy, M. (2011). Spatial Networks. *Physics Reports 499*, 1-101.

Boukreev, K. (2014, January 20). *Genetic Algorithms and the Traveling Salesman Problem*. Retrieved from Code project: http://www.codeproject.com/Articles/1403/Genetic-Algorithms-and-the-Traveling-Salesman-Prob

Brown, G. (2004). Freedom and Transience of Space. In R. Kronenburg, *Transportable Environments 2* (pp. 3-4). London : New York: Spon Press.

Busch, B., Ladurner, G., Baharlou, E., & Menges, A. (2011). Adaptive Sturcture: A Modular System for Generative Architecture. *14th Generative Art Conference*, (pp. 89-91). Roma.

Dalton, R. C., Hölscher, C., & Turner, A. (2012). Understanding Space: the nascent synthesis of cognition and the syntax of spatial morphologies. *Environment and Planning-Part B* , 7.

De Wolf, T., & Holvet, T. (2006). Emergence and Self-Organization: a statement of similarities and differences. *International Workshop on Engineering Self-Organising Applications.* (p. 103). Proceedings of international Workshop on Engineering Self-Organising Applications.

De Wolf, T., & Holvoet, T. (2004). Emergence and Self-Organisation: a statement of similarities and differences. In G. D. Serugendo, *Engineering Self-Organising Systems* (pp. 1-15). Leuven.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 269–271.

Fellous, J., & Arbib, M. A. (2005). *Who needs emotions? the brain meets the robot.* Oxford New York: Oxford University Press.

Fischer, M. M. (1997). Computational neural networks: a new paradigm for spatial analysis. *Environment and Planning A , volume 30*, 1873-1891.

Flake, G. W. (2000). *The computational beauty of nature : computer explorations of fractals, chaos, complex systems, and adaptation.* Cambridge: MIT Press.

Franz, G., Mallot, H. A., & Wiener, J. M. (2005). Graph-based models of space in architecture and. *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics.*, (pp. 1-7). Baden-Baden, Germany.

Freeman, L. (1977). A set of measures of centrality based upon betweenness. *Sociometry 40*, 35-41.

Galanter, P. (2003). What is Generative Art? Complexity Theory as a Context for Art Theory. *6th Generative Art Conference.* New York.

Gross, M. (n.d.). Grids in Design and CAD. *Proceedings ACADIA 91-Reality and Virtual Reality* (pp. 33-43). Los Angeles: ACADIA.

Hensel, M. M. (2010). *Emergent technologies and design.* Oxon: NY: Routledge, 2010.

Hensel, M., Menges, A., & Weinstock, M. (2004). Morphogenetic Design Strategies_Morphogenesis and the mathematics of emergence. *Architectural Design Vol.74* .

Herr, C. (2002). Generative Architectural Design and Complexity Theory. *5th Conference on Generative Art.* Proceedings of the 5th Conference on Generative Art.

Heyligen, F. (2012). *The science of self-organisation and adaptivity.* UNESCO Publishing-Eolss Publishers.

Hillier, B., & Hanson, J. (1984). *The social logic of space* . Cambridge, UK: Cambridge University Press.

Hillier, B., & Hanson, J. (1984). *The Social Logic of Space* . Cambridge: Cambridge University Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. NatL Acad. Sci.* (pp. 2554-2558). Proc. NatL Acad. Sci.

Hübner, W., & Mallot, H. A. (2002). Integration of metric place relations in a landmark graph. *Artificial Neural Networks—ICANN 2002* (pp. 825-830). Springer Berlin Heidelberg: Artificial Neural Networks—ICANN 2002.

Knuth, D. E. (1997). *The art of computer programming.* Mass.: Addison-Wesley.

Krawczyk, R. J. (2002). Architectural Interpretation of Cellular Automata. *Proceedings of 5th Generative Art Conferance.*

Kronenburg, R., Joseph, L., & Wong Y, C. (1989). *Transportable environments 2.* London : New York: Spon Press.

Kuipers, B., Tecuci, D., & Stankiewicz, B. (2003). The skeleton in the cognitive map: a computational and empirical exploration. *Environment and Behavior*, 80–106.

Lynch, K. (1960). *The Image of the City* . Cambridge MA : MIT Press.

March, L., & Steadman, P. (1974). *The Geometry of Environment: An introduction to spatial organization in design.* Cambridge: MIT Press.

Menges, A., & Ahlquist, S. (2011). *Computational Design Thinking.* Chichester, UK: John Wiley & Sons.

Moravec, H., & Elfes, A. (1985). High resolution maps from wide angle sonar. *Proceeding of the IEEE International Conference on Robotics and Automation*, (pp. 116-121). St. Louis.

Müller-Schloer, C. (2004). Organic Computing. *International Conference on Hardware/Software Codesign and System Synthesis* (pp. 8-10). Stockholm: N.Y: Association for Computing Machinery.

Newman, M. (2005). A measure of betweenness centrality based on random walks. 39-54.

Newman, M. (2010). *Networks: An Introduction.* Oxford: Oxford University Press.

Obitko, M. (2014, January 29). *www.obitko.com*. Retrieved from www.obitko.com: http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks 32*, 245.

Proulx, S. R., Promislow, D. E., & Phillips., P. C. (2005). Network thinking in ecology and evolution. *Trends in Ecology and Evolution*, 345-353.

Sabidussi, G. (1966). The centrality index of a graph. . *Psychometrika 31*, 581-603.

Spider Web. (2014, January 20). Retrieved from GBL Web Site: http://www.gbl.tuwien.ac.at/Archiv/digital.html?name=Spider_Web

Terzidis, K. (2006). *Algorithmic Architecture.* Oxford Burlington: MA: Architectural Press.

Thesaurus, C. A. (2014, January 29). *http://dictionary.cambridge.org*. Retrieved from http://dictionary.cambridge.org: http://dictionary.cambridge.org/dictionary/british/network_1?q=network+

van der Zee, A., & de Vries, B. (2008). Design by Computation. *11th Generative Art Conference*, (pp. 35-37). Eindhoven.

Wasserman, S., & Faust, K. (1994). Social Network Analysis in the Social and Behavioral Sciences. In S. Wasserman, & K. Faust, *Social Network Analysis: Methods and Applications* (pp. 1-27). Cambridge University Press.

Yegnanarayana, B. (2004). *Artificial neural networks.* PHI Learning Pvt. Ltd.