Axial Line Placement in Deformed Urban Grids

Final Honours Research Report

George D Konidaris

Supervisor: Ian Sanders

October 29, 2001

School of Computer Science
University of the Witwatersrand
Johannesburg

Abstract

The problem of placing axial lines in configurations of convex, non-overlapping polygons originates in the technique of space syntax analysis, which is used in town planning to describe and analyse architectural structures. Unfortunately, the general problem has been found to be NP-Complete, because of the possibility of configurations in which local choices have to be made, which affect the global optimality of the solution. Because of this, previous research has focused either on finding special cases where an exact solution can be obtained in polynomial time, or heuristic algorithms where approximate solutions can be found in polynomial time.

Recently, it has been conjectured that the exact solutions to a class of convex polygon configurations referred to as deformed urban grids can be found in polynomial time, despite the fact that previous research assumed it to be NP-Complete, and focused on finding approximate solutions for it. Since deformed urban grids are reasonably representative of some actual urban configurations, an algorithm to solve this class of problem exactly would have significant practical application. This report presents some of the properties of deformed urban grids, which would appear to restrict the problem significantly, along with a discussion and demonstration of the choice that still remains within the bounds of the problem. An argument is therefore given that a significant choice factor still exists within the problem, and that a polynomial-time algorithm that solves the problem is unlikely to exist. In addition, an $O(n^4)$ heuristic algorithm based on the derived properties of deformed urban grids is given, as well as a discussion of the relationship between axial line placement in deformed urban grids and other problems, highlighting both the specific problems that a polynomial-time solution would be required to solve, as well as the likely ways in which the problem could be shown to be NP-Complete.

Dedication

This research report is dedicated to a very special group of people (especially Nadia) who, through their intelligence, originality, insight, openness, outlook and sheer sense of joy have made my time at Wits four years of the most deeply enjoyable experiences of my life, and, in their own special way, have contributed greatly towards my education.

Acknowledgments

This report would not have been possible without the support and input of several of my lecturers and fellow students. Firstly, I would like to thank Professor Ian Sanders, for inventing an interesting problem and letting me play with it for a little while. Nadia Uys displayed superhuman patience by letting me ramble on about my research for hours on end when I should have been paying more attention to her. George Christelis, Dylan Scott-Dawkins and Dylan Shell also listened patiently to many an incoherent narrative of my progress, and were responsible for the deaths of several crazy ideas, and the miraculous survival of some even crazier ones. Sheila Rock and Lex Holt have made their presence felt throughout my as of yet infantile research career. Charmaine Venter deserves acknowledgement for providing gentle reminders of long forgotten mathematics, as does Leonard Hagger for being the only fellow student who really knew what a deformed urban grid was. Philip Machanick and Vashti Galpin deserve credit for their tireless attempts to explain what research actually is, and how to write about it. Finally, I am indebted to my parents and my sister, for endeavouring to provide an environment in which I could pace, undisturbed, for as long as necessary.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

The idea of placing axial lines in a set of convex polygons representing an urban space originated in Hillier *et al.* [1983], where a method called *Space Syntax* was proposed to describe and analyse patterns of architectural space. Sanders [2001b] considered the algorithmic problems posed by the space syntax method, and focused on the particular problem of placing axial lines in configurations of convex polygons, placing it in an algorithmic context and abstracting away from the architectural details of the method. Although the axial line placement problem is only one component of the process required to complete the space syntax analysis method [Sanders 2001b], it represents one of the more difficult components of the problem and a general algorithm to solve it would contribute greatly to the automation of the method.

Unfortunately, the general problem of placing axial lines in configurations of convex polygons has been shown to be NP-Complete [Sanders 1999]. Therefore, rather than attempting to exactly solve the more general problem, special cases of the problem were considered in which exact solutions can be found in polynomial time. Recently, it has been conjectured [Sanders 2000] that a solution to the significant set of configurations of convex polygons referred to as *deformed urban grids* is obtainable in polynomial time. Since this class is more general than most previously considered classes, and since it has significant practical application, this research considers the restricted problem of placing axial lines in deformed urban grids, and shows that despite the restrictions that the deformed urban grid class of configurations places on the problem, sufficient choice still remains to make the problem hard, and unlikely to be solvable in polynomial time. A heuristic algorithm, which generates approximate solutions in $O(n^4)$ time, is also given.

The remainder of this chapter provides a brief introduction to the problem, its potential application, the requirements for a solution, and the research method employed. First, section 1.2 outlines the problem of axial line placement in deformed urban grids. Section 1.3 then justifies the proposed research by discussing the importance and potential application of the problem, and is followed by a discussion on the nature of what can be considered to be a solution to the problem in section 1.4. Section 1.5 presents an overview of the research approach used, and an outline of the remainder of this report is given in section 1.6.

1.2 Axial Line Placement in Deformed Urban Grids

Given a set of convex polygons, the axial line placement problem is the problem of placing a set of not necessarily orthogonal "axial" line segments such that all the adjacencies (shared edges) between the polygons are crossed by at least one line, and all of the lines remain entirely within the polygons. In the proposed research, we will consider the variant of this problem where a solution requires the *minimum* set of *maximal* lines – the least number of lines required to cross *all* adjacencies, where each line crosses as many adjacencies as possible, as this realistically reflects the constraints of the original problem [Sanders 2001b].

Deformed urban grids consist of horizontal and vertical corridors, formed by deformed rectangles (where the sides are not required to be of equal length), which meet at intersections, made up of deformed squares. The corridors therefore form a structure similar to but less uniform than a regular grid, which requires the corridors to be rectangular and intersections square. The problem of axial line placement in deformed urban grids therefore refers to the axial line placement problem where the set of convex polygons are constrained to form a deformed urban grid. A more formal and detailed description of both the axial line placement problem and deformed urban grids is given in chapter 2.

1.3 Potential Applications

The primary intended application area of the proposed research is the automation of the axial line placement component of the space syntax method. Since the general problem of axial line placement is NP-Complete [Sanders 2001b], attempts to automate this process that rely on finding exact solutions must restrict their applicability to a subset of the possible polygon configurations where exact solutions are possible. Although not all urban layouts can be represented using deformed urban grids [Sanders 2000], they are capable of representing a great many such layouts, and have more significant potential application than all previously considered special cases. The class of deformed urban grids is capable of representing all configurations that form a chain of rectangles [Sanders *et al.* 2000] and all regular urban grids [Sanders 2000], as well as having significantly more application than configurations that form chains of rectangles [Sanders *et al.* 2000]. Furthermore, outside the context of space syntax, the axial line placement problem for deformed urban grids is applicable to many urban layout and design problems, such as drainage and electrical supply optimisation.

1.4 Definition of a Solution

This research had a slightly unusual definition of a solution, since it could have accepted one of two possible solutions, each providing different insight into the task of placing axial lines in deformed urban grids. The first acceptable solution was a polynomial-time algorithm that performs the task of placing axial lines in a set of convex polygons, and a proof that the set of axial lines generated by this algorithm are a minimal set of maximal lines. A solution of this form would have settled the question of whether or not this specific instance of the axial line placement problem is NP-Complete – if a polynomial-time solution is found, the problem is not NP-Complete (unless $P = NP$, which is highly unlikely) [Garey and Johnson 1979]. Unfortunately, this does not seem to be the case. The second possible solution (and the one presented in this report) consists of a persuasive argument that the "choice" used to prove that the general case is NP-Complete is also present in the case of deformed urban grids, and a polynomial-time algorithm that gives a better approximation to a solution than the current algorithm [Soltész 2000] that performs the same task.

1.5 Overview of Research Approach

The first task of the proposed research was the determination of whether or not the problem contains sufficient "choice" for it to be NP-Complete. After an examination of the properties of deformed urban grids, and the demonstration that a significant amount of choice still exists within them despite the restrictions the configuration class places on the axial line placements that are possible within them, the problem was decided to be NP-Complete (or at least not solvable in polynomial time for the reasons it was conjectured to be). Therefore, the decision making process used to come to this conclusion, including proofs of some of the important properties of deformed urban grids, was developed and used as an argument for the unlikelihood of finding an exact polynomial time algorithm. Thereafter, an algorithm was derived to approximately solve the problem. Following the derivation, the theoretical complexity of the algorithm was analysed, and the characteristics of the approximate solutions generated by the algorithm were examined.

If the problem had appeared to be tractable, then a polynomial-time algorithm to solve the problem would have been derived, its complexity analysed, and a proof of correctness given. Time allowing, an investigation into the optimality of the algorithm would have been conducted.

1.6 Structure of the Report

In the following chapter, previous research that deals with similar problems to those dealt with in the proposed research are examined. The chapter presents the background literature that grounds the axial line placement problem in the method of space syntax, and presents the previous research results and conjectures that have developed the problem and led to the current research area. Finally, it provides an overview of the existing literature that is related to the presented research in terms of methods and concepts used rather than problem definition. Chapter 3 outlines the research method used in the research. It presents the formal research hypothesis that underlies the research, and then outlines the methodology that was employed to evaluate the hypothesis.

Chapter 4 presents a discussion on the properties of deformed urban grids. It focuses on the restrictions that the configuration class places on the types of adjacencies and axial lines that can be used in solving them, and then discusses the issue of whether or not sufficient choice remains to make the problem hard, despite the presence of such restrictions. Chapter 5 then develops a heuristic algorithm that runs in polynomial time, and generates approximate solutions for the axial line placement problem in deformed urban grids. The complexity of the algorithm is analysed and some characterisation of the solutions it is likely to produce is given.

Chapter 6 provides a discussion on the implications of the research results presented. It focuses on the ramifications of the results presented in chapters 4 and 5 – on what would be required of a polynomial time solution, and the possible problems that the axial line placement problem in deformed urban grids could be transformed to as part of a proof of NP-Completeness. Finally, chapter 7 concludes this report by providing an overview of the research results, their significance, and some possible areas where future work may prove fruitful.

Chapter 2

Background and Related Work

2.1 Introduction

This chapter presents an overview of the literature related to the problem of axial line placement in deformed urban grids. The significance of the special case of deformed urban grid configurations in axial line placement is explored, along with the potential applications of such an algorithm and the generality of the case compared to previously solved problems. The work that provides the context for the research described in this document is given, with a view to both demonstrating that the deformed urban grid case is a natural extension to the existing results, and that it poses a problem that is of particular importance and relevance. This is followed by a precise definition of the axial line placement problem for convex polygons, and a presentation of the current definition of the deformed urban grid class of convex polygon configurations, which is shown to be inadequate. Finally, an overview of research that was found to be useful or related to the research presented is given, framing the research in the context of other problems with similar characteristics.

Section 2.2 deals with the significance of the problem, exploring the possible applications of the problem and the ramifications of the complexity of the problem for these areas. Next, section 2.3 details the background to the axial line placement problem, showing the sequence of research steps and significant results that lead to the relevance and importance of the deformed urban grid class of polygon configurations. Section 2.4 then presents a detailed and descriptive definition of the axial line placement problem for deformed urban grids. Section 2.5 reviews some related research results, which provided useful approaches and ideas that were used in the presented research, and section 2.6 summarises.

2.2 Significance of the Problem

Currently, no research exists that deals with the special case of the problem of axial line placement in configurations of convex polygons in which exact solutions can be obtained in polynomial time. Previous research [Sanders *et al.* 2000] focused on special cases using configurations of orthogonal rectangles where the axial lines are required to be orthogonal to the Cartesian axes. Clearly the development of a polynomial time algorithm for a special case of convex polygons with arbitrary axial lines would be a significant result.

In particular, the case of deformed urban grids merits special attention since many town plans can be modeled as deformed urban grids [Sanders 2001b], although not all. An algorithm to give an exact polynomial time solution to the problem or a good approximation to an exact solution in a reasonable amount of time (if an exact solution is unobtainable) would have significant practical applications. Furthermore, the deformed urban grid configuration is more general than both the chains of rectangles configuration [Sanders *et al.* 2000] and the urban grid configuration [Sanders 2000] thus far solved, and has more practical applications than the chains of rectangles configuration [Sanders *et al.* 2000], and therefore represents an important, and wide, problem domain.

2.3 Background

This section introduces the background to the problem of axial line placement at a more detailed level than the introduction given in chapter 1. First, the work leading to the formulation of the axial line placement problem is outlined, starting with the development of the space syntax method. Following that, the chronology of research on the axial line placement problem is given, in order to show how the current problem emerges naturally as an important special case that has not yet been properly dealt with.

2.3.1 Space Syntax

Hillier *et al.* [1983] introduced space syntax analysis as a method of objectively studying the form and function of urban layouts, and attempting to rigorously relate social and environmental variables [Sanders 2001b]. The process involves first producing an urban layout map for a given town or city, then partitioning the open space in this map into a set of convex polygons, placing axial lines within these polygons, and using the number and length of the axial lines to calculate an integration factor for the layout. While the method is currently carried out mostly manually [Sanders 2001b], the automation of some (or all) parts of the method would free professional town planners from a significant amount of mindless labour.

Sanders [2001b] described the process and abstracted the computationally difficult elements from it, ultimately focusing on the axial line placement problem, as the other required components (image processing, partitioning urban layouts, calculating the integration factor) are either well studied, or computationally intractable. This abstraction and selection led to the formulation of the axial line placement problem, with the architectural elements removed from the problem statement.

2.3.2 Axial Line Placement

In one of the first published pieces of research on the axial line placement problem [Sanders *et al.* 1997], the problem of orthogonal axial line placement (where all lines are required to be orthogonal to the Cartesian axes) for configurations of arbitrary rectangles was shown to be NP-Complete by transformation from the vertex cover problem for planar graphs. Later work [Sanders 1999] also showed the more general case of axial line placement in convex polygon configurations to be NP-Complete. The underlying reason for the NP-Completeness of the general problem was that there are configurations where "choice" lines exist – two (or more) lines, of which only one is necessary to cross all the adjacencies in the configuration. In the process of building up a solution, when such choice lines are encountered, it may be possible that in the context of a global solution one of them is better than the other, and leads to a better solution; however, this cannot be known from the examination of a purely local solution.

Following the NP-Completeness proof, further research focused on finding special configurations where exact solutions could be found in polynomial time. Sanders *et al.* [2000] gave polynomial time algorithms for the exact solutions to both chains and trees of rectangles, while Sanders [2000] proved that an exact solution for urban grids can be found in polynomial time. The natural extension to the urban grid case is the deformed urban grid case. However, Soltész [2000] assumed that this case was NP-Complete (without providing any reason for doing so) and therefore produced an approximate solution for it that ran in time complexity $O(n^3)$. Unfortunately this solution does not produce good approximate solutions since it uses a somewhat simplistic method to place lines across intersections. In addition, there has been some work [Sanders 2000; Bukovska 2000] towards polynomial-time algorithms that produce approximate partitions of deformed urban grids.

Recently, however, it has been conjectured [Sanders 2000] that the deformed urban grid class is not, in fact, NP-Complete. Since the class of deformed urban grids is both more general than the urban grid and chains of rectangles classes, and has a much larger potential scope for application than any of the other previous cases, the question of whether or not the class can be solved exactly in polynomial time is a significant one. The presented research is therefore a natural and necessary extension of the existing research.

2.4 Axial Line Placement in Deformed Urban Grids

Axial line placement in deformed urban grids is the special case of the more general problem of axial line placement in convex polygons, applied to configurations of polygons that are known to form deformed urban grids. The following sections, which introduce axial line placement in convex polygons, and present the current definition of the configurations of convex polygons which may be labelled deformed urban grids, are based on Sanders [2001b].

2.4.1 Axial Line Placement in Convex Polygons

The axial line placement problem for convex polygons is: given some configuration of convex polygons in the plane, place a set of line segments (known as "axial lines") such that each adjacency (shared edge segment) is crossed by at least one line, and that all of the lines lie entirely within the boundaries of the polygons. In addition, this research will add the requirement that the minimal set of maximal lines (in the sense that each line crosses as many adjacencies as possible) is required. Figure 2.1 shows a configuration of convex polygons; and Figure 2.2 shows the resulting configuration with the axial lines drawn in bold.

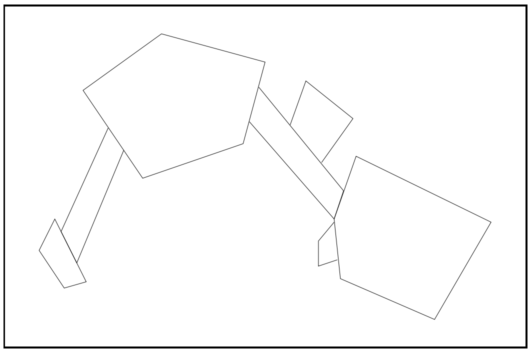

Figure 2.1: A Configuration of Convex Polygons

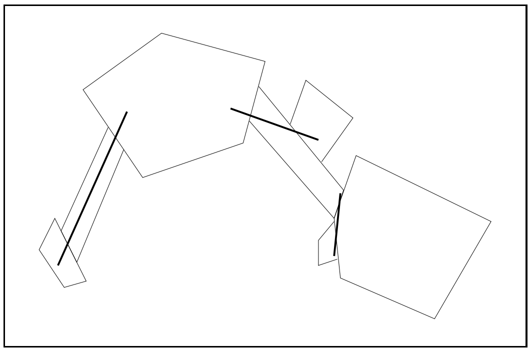

Figure 2.2: A Configuration of Convex Polygons with Axial Lines

The axial lines can be thought of as lines of sight; then, if we consider the adjacencies between polygons to be "doors", and the polygons themselves rooms, we can consider the problem to be the placement of the least number of "guards" (where each guard can see along a single ray only) such that every door is guarded, with the extra provision that each guard watch as many doors as possible. This formulation of the problem led to it originally being known as "ray guarding" [Sanders *et al.* 1997].

In the context of space syntax analysis, the *space between buildings* is of interest, rather than the buildings themselves. Rather than model the buildings as convex polygons, the free space in a town map is modeled as a convex polygon configuration. The axial line placement then gives an indication of how far through a city or town an individual can walk without changing direction, and therefore provides information about the visibility and accessibility of the urban layout. Further details of the space syntax method may be found in Hillier [1996].

2.4.2 Deformed Urban Grids

A *deformed urban grid* is a generalisation of an urban grid, which in turn is a generalisation of a complete urban grid. These concepts will therefore be presented in turn. As they become more general, the concepts will likewise be able to describe a larger subset of all urban layouts, and hence become more applicable.

Definition 2.1 A *complete urban grid* of size n is a complete two-dimensional grid of size n where all grid edges (corridors) have some length l, and width w, such that $w < l$ and all grid vertices (intersections) are of size $w \times w$ [Sanders 2001b].

Figure 2.3 shows a complete urban grid of size 4. The applicability of this class of configurations of convex polygons to town plans is clear when you consider the holes in the grid (the dark squares) to be buildings, and the corridors to be roads, with each intersection forming a four-way stop or traffic light intersection. This type of configuration, however, requires that the town plan be square, a constraint that is extremely unlikely in real urban layouts, which are evolved entities that are often shaped by geographic features. Therefore, the natural way to proceed to make the configuration more general would be to handle subgraphs of complete urban grids.

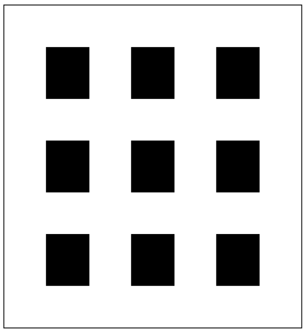

Figure 2.3: A Complete Urban Grid of Size 4

Definition 2.2 An *urban grid* is any subgraph of a complete urban grid [Sanders 2001b].

Figure 2.4 shows an urban grid of size 4. This configuration class is reasonably general and is capable of representing some classic urban layouts, such as the Manhattan borough of New York. However, in order to maximally generalise this class, it would be useful to be able to represent urban layouts where the street and intersection boundaries are not exactly orthogonal. Therefore, the concept of a *deformed urban grid* is introduced.

Definition 2.3 A *deformed urban grid* is an urban grid where the corridors are deformed rectangles (quadrilaterals with two sides which are long compared to the other two sides) and the corridor intersections are deformed squares (quadrilaterals where all of the sides are of roughly similar length). Also the end of a corridor must have the same size and location as the corresponding side of the intersection into which it runs [Sanders 2000].

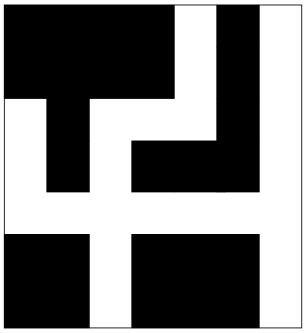

Figure 2.4: An Urban Grid of Size 4

Unfortunately, a new definition is necessary, since this definition can result in an extremely deformed grid which is no longer recognisable as a grid – and no longer possesses the properties of grids which may allow for a polynomial time solution. This can occur because the definition allows for virtually unbounded deformation of the original urban grid, and can lead to the thoroughfares of the grid becoming adjacent to each other. Consider Figure 2.5.

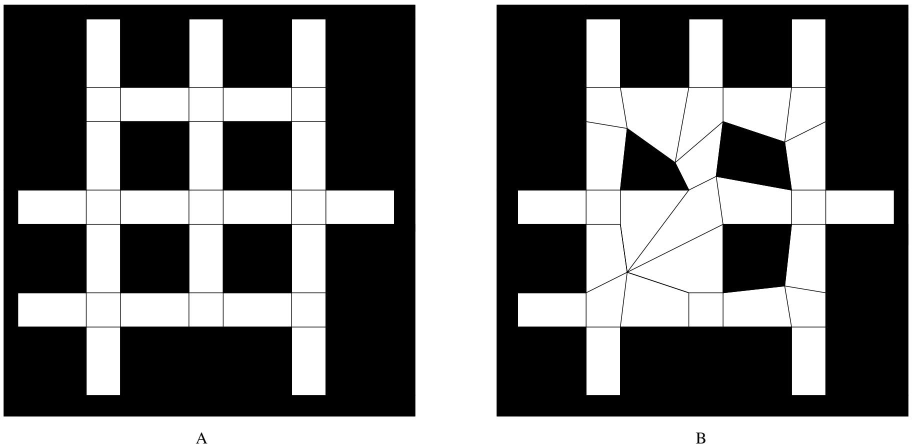

Figure 2.5: An Urban Grid, Deformed Under the Old Definition

Figure 2.5 shows an urban grid A, and a resulting deformation B that is legal under Definition 2.3. The lower left corner of B shows that it is possible under Definition 2.3 for the grid to deform to the point where the boundaries between grid lines are no longer present, effectively allowing for the formation of highly irregular "blobs" made up of intersections and thoroughfares. These will have to be partitioned into some set of convex polygons, and the resulting formation is more likely to resemble an arbitrary configuration of convex polygons than a grid–like configuration of convex polygons.

Figure 2.6 shows the intended form of a deformed urban grid of size 4 [Sanders 2001a]. The irregular shape of the dark areas (which are meant to represent non-space) is more realistic, since it does not assume that urban structures, such as parks, buildings, and museums, have orthogonal boundaries. Even relatively irregular structures can be presented using a deformed urban grid, by using multiple "blocks" to represent a single building, removing the restriction than building boundaries should be straight lines. Despite this, the deformed urban grid class, as originally intended, retains an essentially grid-like structure. A new definition for the deformed urban grid class, which more accurately captures the intended form of deformed urban grids, will be given in chapter 3.

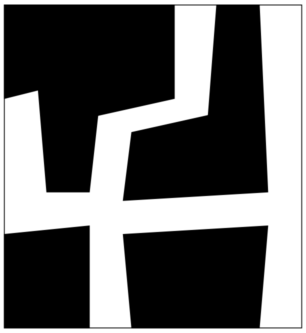

Figure 2.6: A Deformed Urban Grid of Size 4

2.5 Related Research

This section presents research which, although not directly related to the axial line placement problem for deformed urban grids, either contains results that were of use in the construction of a solution for the presented research or general solution methods that were useful in the presented research.

2.5.1 General Methods

Basic algorithmic methods and standard well-known data structures such as interval trees formed the initial basis for the attempted derivation for a solution in the presented research. Additionally, standard complexity analysis and proof methods were employed when analysing derived solutions. Good introductions to these techniques may be found in any introductory algorithmic textbook, such as Baase and Van Gelder [2000] or Brassard and Bratley [1996].

In addition, there are ideas and techniques that are specific to the field of computational geometry but can be widely applied within it, such as the line sweep method and convex hull algorithms. An overview of such methods may be found in O'Rourke [1995].

2.5.2 Visibility

A related problem of significance to axial line placement in convex polygons is that of *visibility*. In the simplest case, given a point a within a convex polygon, any other point b, also in the polygon, is considered visible from a if a line can be drawn between a and b that remains wholly contained within the polygon. This form of visibility is known as *point to point* visibility, and Figure 2.7 shows a polygon, with the set of all points visible from point a shaded. El Gindy and Avis [1981] presents a linear time algorithm for generating the *visibility polygon* for a polygon from a point, a. The visibility polygon is the polygon containing all the points visible from a, and the boundary of the shaded area in Figure 2.7 constitutes the visibility polygon of the outer polygon and a.

More complicated forms of visibility exist, and are surveyed in O'Rourke [1997]. However, of particular importance to this research was the notion of *edge to edge* visibility. Consider Figure 2.8.

In Figure 2.8, the polygon configuration marked A is a corridor as we can expect to find in a deformed urban grid configuration. The polygon marked B is the *adjacency polygon* of A, the boundary of the extreme adjacencies of the corridor. If edge a is visible from edge b, then we can place a single axial line through all of the adjacencies between them. An algorithm to efficiently determine edge to edge visibility therefore has significant potential application to any axial line placement research, and was used in the presented research.

Fortunately, the problem of edge to edge visibility has been well studied, using as a base the linear algorithm [Avis and Toussaint 1981] that determines the visibility of an entire polygon from an edge. Du Plessis and Sanders [2000] presented a linear (and therefore optimal) algorithm for partial edge visibility

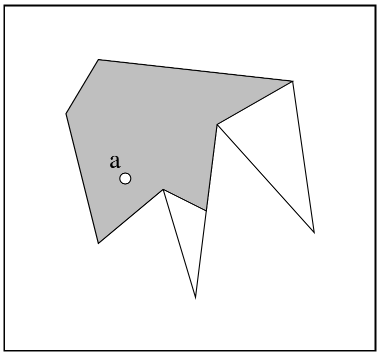

Figure 2.7: A Visibility Polygon from a Point

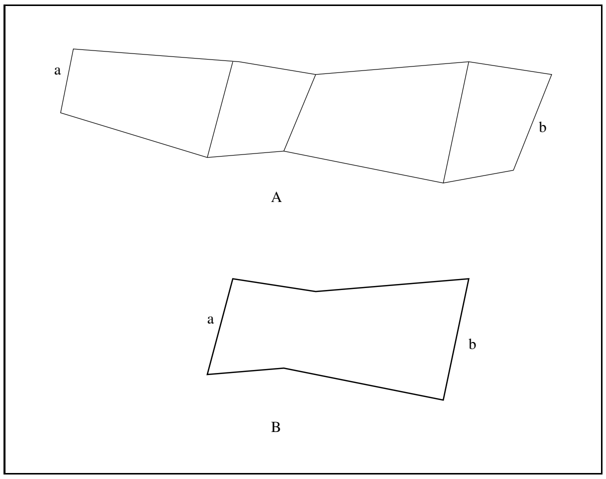

Figure 2.8: The Adjacency Polygon of a Corridor

in chains of orthogonal rectangles, and this has been generalised [Soltész 2000] to handle partial edge visibility in chains of convex polygons in linear time.

2.5.3 Art Gallery Problems

The class of problems widely referred to as *Art Gallery Problems* deals with the minimum number of point "guards" required to be placed inside a polygon (the "art gallery") so that every point in the polygon is visible to at least one point guard. There are several variations on this problem, including both cases where the guards are constrained in terms of their "vision", and cases where the polygon is constrained in terms of its form. Clearly, the way in which both types of constraints can be used to generate efficient solutions provides an analogous approach to the proposed research. O'Rourke [1987] provides an overview of this area.

2.5.4 Finding Adjacencies in Polygon Configurations

In order to place axial lines across the adjacencies between convex polygons, all the adjacencies between the set of polygons are required to be found in a reasonable time. An algorithm to detect the adjacencies between a set of non-overlapping convex polygons is given in Adler *et al.* [2001], that runs in time complexity $\Theta(z \log n)$, where z is the total number of edges in the polygon configuration, and n is the total number of polygons. Although this research assumed that the adjacencies between the convex polygons used to partition the deformed urban grid are already known, this algorithm could be used to determine the adjacency relations between a set of polygons in an actual implementation.

2.5.5 NP-Completeness and Special Cases

In order to critically examine the original NP-Completeness proof for axial line placement in convex polygons, a strong basis for both the concepts and ideas behind NP-Completeness was required, as well as some idea of when special cases allow polynomial time solutions to be found. A brief overview of NP-Completeness can be found in Brassard and Bratley [1996], and was used as a starting point for further investigation. Also of relevance was Garey and Johnson [1979], which contains a thorough treatment of NP-Completeness as well as information on special cases of NP-Complete problems which can be solved exactly in polynomial time. The transformation from biconnected planar vertex cover to axial line placement in configurations of orthogonal rectangles can be found in Sanders [2000], and the proof that biconnected planar vertex cover is NP-Complete can be found in Sanders *et al.* [1997].

By examining the characteristics of the three special cases of axial line placement that have been shown to be solvable in polynomial time [Sanders *et al.* 2000; Sanders 2000] significant insight was obtained as to the nature of the properties that allow exact solutions to be efficiently obtainable. This insight forms the basis of the argument that the problem is NP-Complete, which is given in chapter 4.

2.5.6 Edge Cover

The axial line placement problem can be considered a generalisation of the *edge cover* problem. The edge cover problem is the problem of finding the minimum set of edges in a graph such that every node in the graph is adjacent to at least one edge. The standard edge cover problem can be solved in polynomial time [Garey and Johnson 1979].

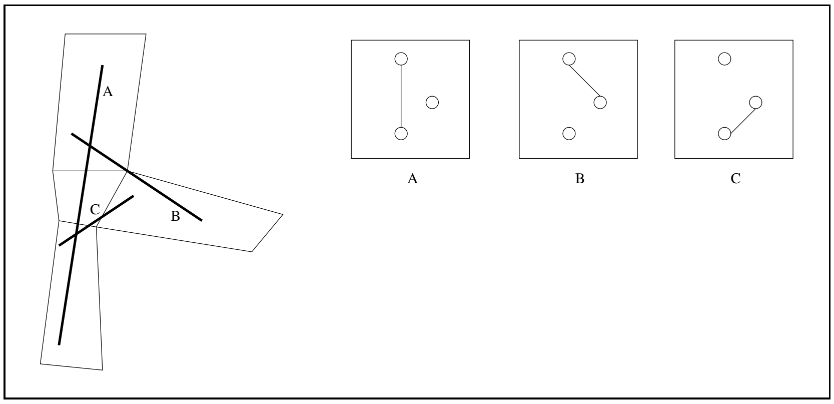

Figure 2.9: Axial Line Placement Viewed as Edge Cover

Figure 2.9 is an axial line placement problem transformed to an edge cover problem, except that rather than requiring individual edges, we are required to find *the minimum subset of the available graphs such that they form an edge cover*. This formulation leads directly to the minimum cover problem.

2.5.7 Minimum Cover

The *minimum cover* problem is the problem of finding a minimum subset M of a set $C = \{c_1...c_n\}$, each of size $s_i, i = 1...n$, such that each of the elements in a third set, P, is in at least one of the sets in M. Unfortunately, this problem is NP-Complete, even when $s_i \leq 3\ \forall i$ [Garey and Johnson 1979]. If each axial line is viewed as a set of adjacencies that it cuts, then the required solution is the minimum set of those sets that contains all of the adjacencies in a graph. While a transformation from minimum cover to axial line placement in deformed urban grid has not been found (and, if found, would prove that axial line placement in deformed urban grids is NP-Complete), the transformation from axial line placement to minimum cover offers significant potential insight into the problem. Therefore, a careful examination of the minimum cover problem and the special cases in which it can be solved in polynomial time was necessary in order to provide insight into the cases in which the axial line placement problem could be solved in polynomial time.

2.6 Conclusion

This chapter has presented the literature related to problem of axial line placement in deformed urban grids. The practical and theoretical significance of the problem was discussed, followed by an overview of the background research that led to the identification of the configuration of deformed urban grids as an important and potentially solvable special case of the more general problem, and finally related research approaches were examined with a view of extracting problem characteristics and methods that could be of use in this research. The following chapter presents an overview of the research method that was adopted in an attempt to resolve the research question, based on the related approaches and background work presented in this chapter.

Chapter 3

Research Method

3.1 Introduction

In order to critically examine the question of whether or not the axial line placement problem in deformed urban grids can be solved in polynomial time, a formal hypothesis around which the research is to focus should be established, and a methodical approach to accepting or rejecting that hypothesis should be decided upon. This chapter therefore provides an outline of the research methodology that was used to conduct the research into axial line placement in deformed urban grids. It provides a basis for the remainder of the research, describing the research hypothesis that underlies the research as well as the methods used and steps taken to reject that hypothesis. It also introduces a new definition for the deformed urban grids class of convex polygon configurations, which was used throughout the research.

The chapter begins by stating the formal research hypothesis in section 3.2, which formed the focal point of this research. Section 3.3 then presents the new definition of the deformed urban grids class, and the approach that was adopted in order to reject the hypothesis, as well as the approach that would have been used to accept it. Section 3.4 outlines the sequence of stages through which the research proceeded, and section 3.5 provides a discussion which aims to show that the steps taken and information gathered are sufficient to reject the hypothesis. Finally, section 3.6 provides a summary of the material covered in this chapter.

3.2 Statement of Hypothesis

The formal research hypothesis of the this research was:

> *Given a configuration of convex polygons that form a deformed urban grid, it is possible to find the minimum number of maximal arbitrary line segments entirely contained within the polygons, such that every adjacency is crossed at least once, in polynomial time.*

The following sections detail the approach taken and strategy used to reject the research hypothesis, and a discussion of whether or not the course of action taken was sufficient to do so.

3.3 Approach

The approach adopted in this research to provide insight into the problem involved providing either an existence proof of a polynomial-time solution to the problem of axial line placement or a compelling argument of why no such solution could exist. Since a new definition of the deformed urban grid class was required for this research (as explained in chapter 2), the following definition, which more accurately captured the original intention of the class [Sanders 2001a], was used:

Definition 3.1 A *deformed urban grid* is any structure that can be obtained from an urban grid, where each intersection is deformed *inwardly* (meaning that the deformed intersection must fall entirely within the boundaries of the original intersection), and required to remain convex, and the endpoints of each corridor must remain attached to the endpoints of the intersections they were originally attached to.

The new definition allowed for the examination of the problem rigorously; since the class is now well-defined, any properties observed or required could now be conclusively proved or disproved. This provided a basis for the determination of the restrictions that deformed urban grids place on their potential convex polygon partitions and axial line placements.

The primary method of determining the complexity of the problem was to examine the issue of choice. The conjecture that the axial line placement problem in deformed urban grids is solvable in polynomial time [Sanders 2000] was based upon the perception that the convex polygons used to partition the deformed urban grid would be aligned along thoroughfares, and that the axial lines would also therefore be aligned along thoroughfares, and the choice involved between such a set of axial lines would therefore be constrained. While this research supports the first perception, it demonstrates that the second does not hold. If it had held, then a polynomial-time algorithm would have been derived using the insights gained while examining the issue of choice.

3.4 Strategy

The first step in the proposed research was the pivotal NP-Completeness decision – in order to decide upon which path to take, this decision needed to be made carefully, and backed up by a solid argument. Furthermore, when examining the aspects of choice that lead to the NP-Completeness of the original problem, and the aspects of the deformed urban grid configuration that both allow and limit it, sufficient insight into the problem to construct a good approximate solution was gained.

Should the problem have appeared to be tractable, then a typical solution strategy would have been used: the problem would have been examined from a few different viewpoints, and an algorithm would have been developed, using the standard computational geometry methods as a starting point. This algorithm would in all likelihood have been centered around the intersection portion of the deformed urban grid configurations – this is the portion where all the choices are made, and the portion where solutions to simple sub-cases will have to be combined. Following the derivation of the algorithm, a complexity analysis would have been performed, followed by a proof of correctness. Finally, if time had allowed, an investigation into the optimality of the algorithm would have been conducted.

However, since the problem appears to be NP-Complete, an algorithm was developed which took advantage of some of the properties of deformed urban grids (uncovered during the NP-Completeness decision) in order to provide an efficient approximation algorithm which produces good approximations to the exact solution in most cases. Following the derivation of the algorithm, a complexity analysis was performed, and a brief characterisation of the types of solutions generated by it was given.

Either variation on the proposed research would have provided a useful result in that it would have either provided an exact solution to the problem, or (as was the case) an argument as to why no efficient solution could exist, along with an accompanying approximate solution algorithm.

3.5 Conclusively Determining Complexity

In order for the presented research to be effective, it is required to be able to provide either an answer, or strong supporting evidence for either accepting or rejecting the research hypothesis – effectively, it is required to be able to *answer the research question*, and show why (or why not) a polynomial time algorithm for axial line placement in deformed urban grids exists.

Clearly, the existence of a polynomial-time algorithm that is proven to solve the problem of axial line placement in deformed urban grids would have allowed us to accept the hypothesis. Provided a proof method could have established conclusively that the algorithm works for all cases, it would have constituted a full and final existence proof that the hypothesis is correct.

Rejection of the hypothesis is not as clear cut. Firstly, an "argument" as to whether or not the NP-Completeness proof still holds is somewhat less than mathematically rigorous. However, since the pivotal feature of the proof is the presence of "choice lines", a sufficiently convincing demonstration of the continued existence of these lines means that the proof is still likely to hold. In addition, since $P \neq NP$ has not been proven to be true, or shown to be false, even the NP-Completeness proof holding may not necessarily prove that the problem cannot be solved in polynomial time – even though it is widely suspected that $P \neq NP$, to the extent that it is axiomatic to much current research. However, either way, a significant statement is made about the difficulty of the problem, and if the choice issue remains pivotal to the problem, then the hypothesis can be considered to be rejected *within the context of an Honours-level research report*.

3.6 Summary

This chapter presented the formal research hypothesis that the given research aimed to accept or reject. It also presented the approach that was used to do so, and a strategy that outlined the steps performed to implement the research. Finally, a discussion was given in which the validity of the approach adopted to answering the research question was addressed. The following chapters present the results of the application of the research methodology presented in this chapter, including both the demonstration of sufficient choice to reject the hypothesis, and an approximate algorithm to solve the axial line placement problem for deformed urban grids.

Chapter 4

Properties of Deformed Urban Grids

4.1 Introduction

Before any attempt to derive an algorithm for placing axial lines in deformed urban grids is made, it is important to understand the properties of both deformed urban grids and the axial lines that can occur within them. Since this research adopts a new definition for deformed urban grids, and since previous research [Sanders 2000; Soltész 2000] did not discuss their properties to any level of depth, an investigation into the properties of deformed urban grids is necessary.

This chapter therefore presents some of the properties of deformed urban grids that can be derived from the new definition, and are useful in the context of axial line placement. Of particular interest is the *Thoroughfare Theorem*, which provides useful insight into both the nature of the convex polygons that can be used to cover the deformed urban grid, and the degree to which an axial line can cross multiple thoroughfares. The consequences of the thoroughfare theorem are then examined, and both the types of adjacencies that could be present in a deformed urban grid configuration and the types of axial lines allowable within such a configuration are enumerated. The special case where all the axial lines are required to be orthogonal to one of the Cartesian axes is then briefly discussed. This is followed by an examination of the issue of choice within axial line placement configurations in deformed urban grids, which demonstrates that despite the restrictions that the Thoroughfare Theorem places on such configurations, sufficient choice remains to render the problem unlikely to be solvable in polynomial time.

Section 4.2 introduces and defines some of the new terminology that will be used for the remainder of this document. Section 4.3 introduces and proves the Thoroughfare Theorem and four of its corollaries, after which section 4.4 presents the possible forms of adjacencies, and section 4.5 presents the possible line placements that can be generated for deformed urban grids. Section 4.6 examines the effect of the requirement that the axial lines be orthogonal has on the difficulty of the problem. Section 4.7 discusses the critical issue of choice in axial line configurations in deformed urban grids, and section 4.8 summarises.

4.2 Terminology

The following terms will be used for the remainder of this chapter, and throughout the rest of this document:

Definition 4.1 A *thoroughfare* in an urban grid is a chain of alternating rectangles (size $w \times l$) and squares (or intersections, size $w \times w$) running vertically or horizontally. In a deformed urban grid, a thoroughfare is the structure that resulted from the deformation of a thoroughfare in an urban grid.

Definition 4.2 A *bridge* is any rectangle (of size $w \times l$) that joins two thoroughfares running in parallel in an urban grid. In a deformed urban grid, a bridge is the quadrilateral structure resulting from the deformation of a bridge in an urban grid.

Definition 4.3 Two thoroughfares in an urban grid are termed *orthogonal* if they do not run in parallel. In a deformed urban grid, they are orthogonal if they resulted from the deformation of two orthogonal thoroughfares in the urban grid.

Definition 4.4 *Intersection adjacencies* are adjacencies that are defined along the border of an intersection.

Definition 4.5 *Diagonal intersection adjacencies* are adjacencies that are defined inside an intersection, but fall diagonally across it, rather than along its border.

Definition 4.6 *Essential lines* are axial lines that are the only lines to cross at least one adjacency within a deformed urban grid. Any line that is marked essential must be included in a solution set of axial lines, since leaving it out would leave an adjacency uncrossed.

Definition 4.7 An adjacency is termed *owned* when it is crossed by an essential line. Such an adjacency can never cause another axial line to be part of a solution because it has been crossed at least once by a line already in the solution.

Definition 4.8 The *total length* of an axial line is the number of adjacencies it crosses.

Definition 4.9 The *unowned length* of an axial line is the number of unowned adjacencies it crosses.

The definitions given above will form the basis for the discussion of the properties of deformed urban grids for the remainder of this chapter, and the remainder of this report.

4.3 The Thoroughfare Theorem

Consider Figure 4.1, which depicts an intersection in an urban grid (where $w < l$), and the adjacent rectangles of its surrounding thoroughfares.

Theorem 1 ***(The Thoroughfare Theorem)*** *In Figure 4.1, $\gamma > \pi$, and is therefore reflex.*

Proof

First, it is clear that $\alpha = \beta$, since $\Delta AEB \equiv \Delta CDB$ (all sides equal).

Further, since $tan(\alpha) = \frac{w}{l}$:

$$\begin{aligned}
&w < l \Rightarrow \tfrac{w}{l} < 1 \\
&\Rightarrow tan(\alpha) < 1 \\
&\Rightarrow \alpha < \arctan(1) \\
&\Rightarrow \alpha < \tfrac{\pi}{4} \\
&\Rightarrow \beta < \tfrac{\pi}{4} \\
&\Rightarrow \alpha + \beta < \tfrac{\pi}{2}
\end{aligned}$$

But, since γ, α, β and $\frac{\pi}{2}$ are angles around a point:

$$\begin{aligned}
&\gamma = 2\pi - (\tfrac{\pi}{2} + \alpha + \gamma) \\
&\Rightarrow \gamma = \tfrac{3\pi}{2} - (\alpha + \beta) \\
&\Rightarrow \gamma > \tfrac{3\pi}{2} - \tfrac{\pi}{2} \\
&\Rightarrow \gamma > \pi \qquad \square
\end{aligned}$$

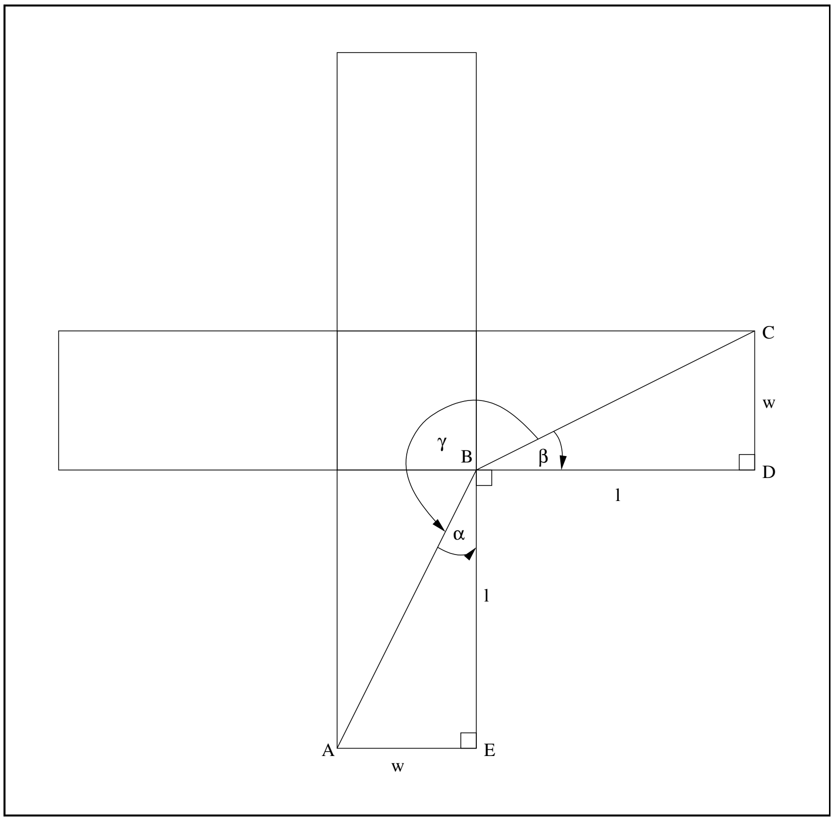

Figure 4.1: An Intersection in an Urban Grid

This theorem holds for all such angles between two orthogonal thoroughfares, as can be seen by simple rotation or reflection of Figure 4.1. In addition, the following corollaries follow directly from the Thoroughfare Theorem:

Corollary 1 *The angle γ remains reflex when the urban grid is subjected to deformation leading to a deformed urban grid.*

This follows because the points A and C are constrained by the deformation process to move further inside their intersections, or stay where they are. Clearly, if either stays where they are, the other moving inside the intersection can only make γ greater. The same argument holds for both of them moving within their intersections.

Corollary 2 *The outer angles formed by the lines A–B–D, E–B–D and E–B–C are all reflex, and remain so when the urban grid is subjected to deformation leading to a deformed urban grid.*

This follows because the deformation leading to an urban grid does not allow any of the points to leave the original intersection. Therefore, the outer angle formed by the line A–B–D must be reflex, since it is larger than that originally formed by A–B–C, which is also reflex. A similar argument holds for the angles formed by E–B–D and E–B–C.

Finally, and most importantly, the Thoroughfare Theorem and the two corollaries above lead to the following corollaries that are directly applicable to the axial line placement problem in deformed urban grids:

Corollary 3 *A convex polygon used in the partitioning of a deformed urban grid must be a subset of a thoroughfare.*

Proof (by contradiction)

Assume that a convex polygon covers space in two distinct thoroughfares. Clearly, it cannot cover space in two non-orthogonal thoroughfares without having crossed into the second one over a bridge. Therefore, it must at least cross one thoroughfare that is orthogonal to its first thoroughfare.

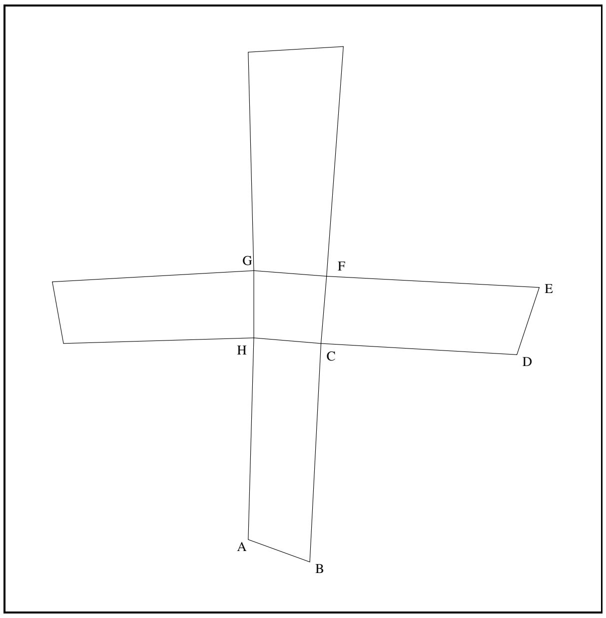

Figure 4.2: A Vertical and Adjacent Horizontal Thoroughfare

Figure 4.2 depicts a vertical thoroughfare and adjacent horizontal thoroughfare. In order for the polygon to cover parts of both thoroughfares (and since the polygon's points must line on the corners of a deformed rectangle or deformed square), it must have at least one of A and B (we will call this K), and at least one of D and E (we will call this L), as vertices. Then, since a polygon is a closed path, there must be exactly two paths from K to L, a *right path* and a *left path*. We will consider the right path.

The right path clearly cannot be K–C–L, since we have shown that that line forms a reflex angle, and hence cannot be part of a chain that forms a convex polygon. It also cannot be K–F–L, since that line does not allow the left chain to reach L without creating a non-simple polygon. It also cannot start with the lines K–H or K–G, since that also forces the left chain to form a non-simple polygon. Finally, it cannot extend past the G–F line, since again the left chain cannot proceed to L without crossing the right chain. Hence, no such right chain can exist – contradiction. □

Corollary 4 *No axial line can be placed such that it cuts adjacencies in the polygons associated with more than one thoroughfare and its intersections, and perhaps in one bridge at either end of the line.*

Consider the axial line L shown in Figure 4.3. We will assume it is associated with the shaded thoroughfare in Figure 4.3, and that it crosses the line A–B (which may or may not be an adjacency). If it does not cut A–B, it either cuts the C–D line, in which case we could consider it rather to be associated with the unshaded horizontal thoroughfare, or it does not, in which case we could consider either of the deformed rectangles it lies within to be the bridge on its end, and the remaining deformed rectangle and intersection to be the thoroughfare it is associated with.

It simply remains to observe that the line L could never cut the adjacency line C–D since in order to do so it must "bend" at E. The line A–E–C has already been shown to be reflex (and hence not a straight line), so the only direct line that can extend from A (or B) to C (or D), must pass below E, and hence fall outside of the deformed urban grid, which it not allowed.

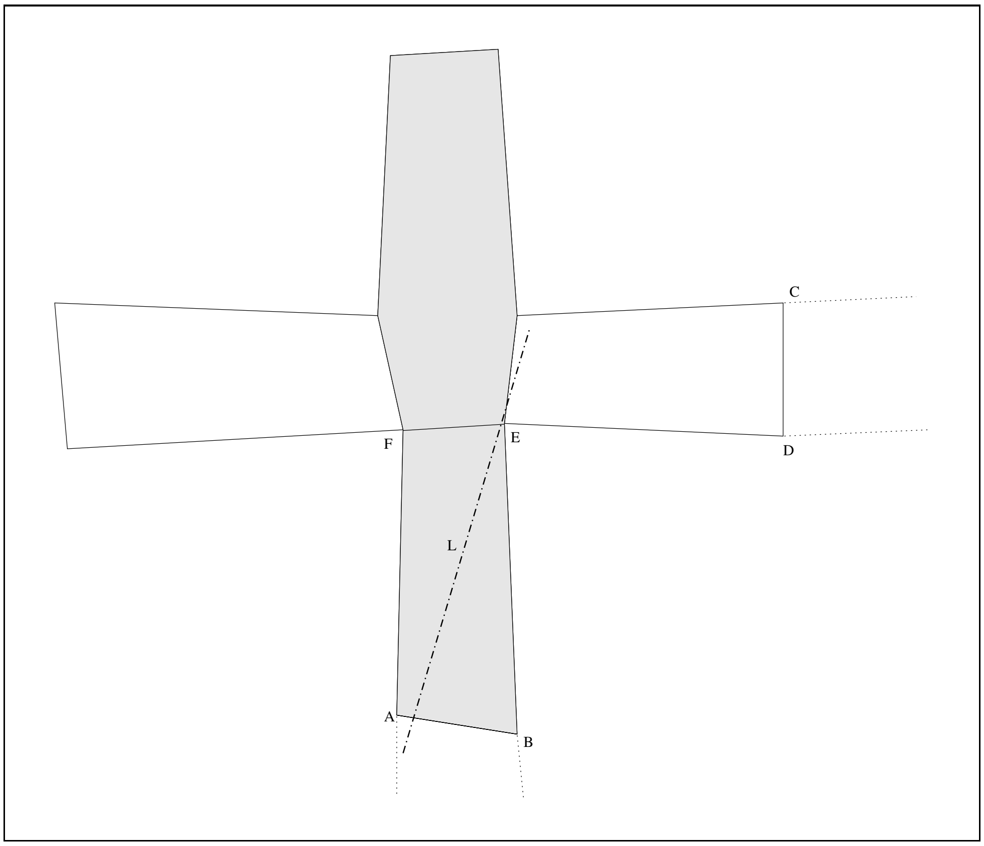

Figure 4.3: An Axial Line in a Deformed Urban Grid

The Thoroughfare Theorem and the four corollaries that follow directly from it allow for the characterisation of both the types of adjacencies that can be found in deformed urban grids and the types of axial lines that can be placed to cross them. The following two sections will enumerate the possible forms of both of these.

4.4 Potential Adjacencies

As is the case with urban grids, the deformed urban grid class of configurations allows for only a fixed number of types of adjacencies to be created when a configuration is partitioned. However, unlike the case of urban grids, they do not usefully restrict the placement of axial lines. There are two types of potential adjacency configurations: intersection configurations, where all of the adjacency lines are placed on or within an adjacency, and thoroughfare configurations, where all of the adjacency lines are placed on or within a thoroughfare, but not necessarily within an intersection. Figure 4.4 presents the intersection configurations that are possible within a deformed urban grid. Note that all adjacency types are given vertically, and may be flipped (reflected about the x-axis) or mirrored (reflected about the y-axis).

All of the adjacencies marked in Figure 4.4 are conditional upon their adjoining bridge's presence; for example, in configuration A, the left adjacency will not be present if there is no adjoining left bridge. Of the adjacency types in Figure 4.4, only C cannot occur within an urban grid configuration. However, none of the adjacency configurations shown in Figure 4.5, which presents the thoroughfare configurations that are possible within a deformed urban grid, can occur within an urban grid. Note that the adjacency lines in Figure 4.5 that do not terminate on an intersection may extend as far as the thoroughfare they are in does; a particular polygon boundary line might constitute a line of type G on one end, F on the other (mirrored and flipped), and cross several intersections in a configuration of type H, along the way.

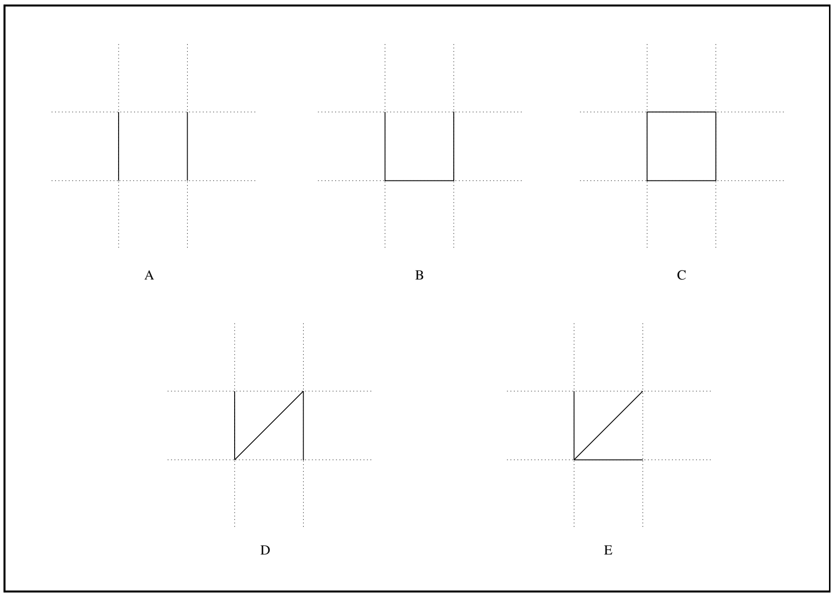

Figure 4.4: Possible Intersection Configurations of Adjacencies

In addition, type F allows for any of the intersection types to be present in the intersection below it, provided the thoroughfare that is is contained in is "sealed" from the other thoroughfare by one of the lines in the intersection adjacency configuration.

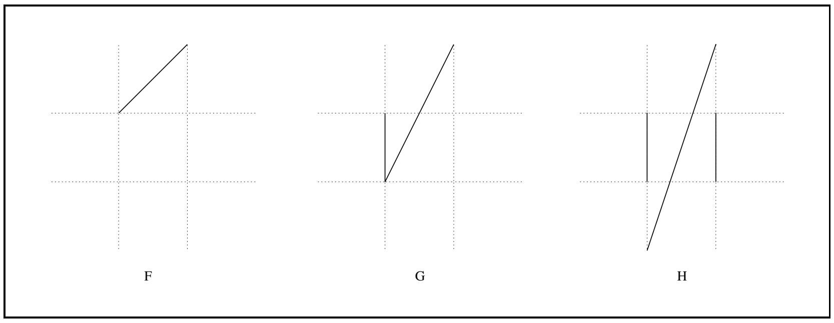

Figure 4.5: Possible Thoroughfare Configurations of Adjacencies

The thoroughfare adjacency configurations offer significant insight into the difficulty of placing axial lines in deformed urban grids, since they demonstrate quite clearly that the types of adjacencies do not significantly limit the types of lines, nor do they limit the levels to which those lines can cross adjacencies, as they do in the urban grid case. It is clear from even a superficial examination of these lines (for example, type H), that a single adjacency may be crossed not only by lines lie along its thoroughfare, but also by any line going through any of its intersections – including lines that lie along orthogonal thoroughfares.

4.5 Potential Line Placements

The conjecture that axial lines can be optimally placed in the deformed urban grid class of convex polygon configurations was based on the insight that the polygons used to partition the deformed urban grids would lie along thoroughfares [Sanders 2000]. This has been shown through the Thoroughfare Theorem. However, the conjecture also stated that the axial lines would then be placed so that they were aligned along those thoroughfares. While the properties of deformed urban grid configurations severely restrict the form of allowable axial lines, there is still a significant amount of variation in the types of line. Corollary 2.4 provides insight into the allowable variations in line placement, which are depicted in Figure 4.6.

Note that each of the types of lines in Figure 4.6 may also be aligned horizontally, flipped or mirrored. Also note that while the boundaries of the deformed quadrilaterals and squares are shown, they may or may not actually be adjacency lines, depending on how the configuration is partitioned. There may also be adjacency lines within the deformed quadrilaterals and squares, again depending on the partition.

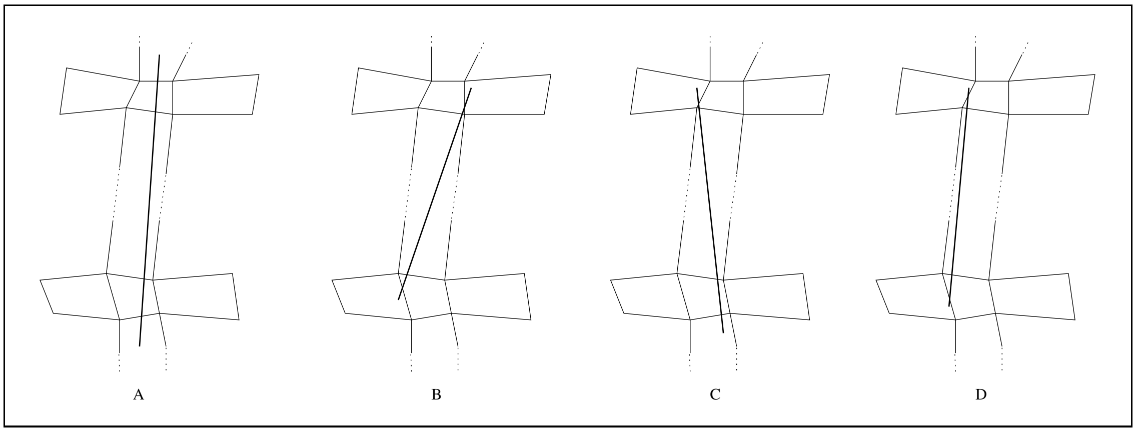

Figure 4.6: Allowable (Vertical) Axial Line Types

The line labelled type labelled A is a *thoroughfare line*, which is strictly aligned along a thoroughfare. These types of lines are likely to cover long, straight thoroughfares, but unlike the urban grids case, more than one line may be required to cover a single thoroughfare. The line type labelled B is a *diagonal line*, which covers adjacencies along a single thoroughfare, but also inside a bridge on either end of the line. The line type labelled C is a *half-diagonal line*, which only cuts adjacencies in a bridge on one side of the line. Finally, the line type labelled D is a special case of a diagonal line called a *side line* which covers bridges on both sides of the line, but both bridges are on the same side of the associated thoroughfare.

4.6 Orthogonal Axial Lines

Previous work [Sanders *et al.* 2000; Sanders 2001b; Sanders *et al.* 1997] on axial line placement has often focused around the special case of orthogonal axial line placement – where all of the axial lines are required to be orthogonal to one of the Cartesian axes. This case simplifies the problem significantly, because it restricts the potential axial line placements allowed by removing a degree of freedom from them.

Unfortunately, in this case, the restriction does not appear to aid solving the problem significantly. The primary reason for this is that the deformation of the grid allows for it to be deformed such that the advantage of the removal of the degree of freedom from the axial line placement can be nullified by the extra degree of freedom inherent in the deformation of the grid. The grid can be "bent" in such a way as to remove virtually all of the advantages of forcing the axial lines to be orthogonal to the axes.

However, the orthogonality restriction *does* simplify the problem somewhat – for example, adjacency types A and B cannot occur within the same thoroughfare segment provided the type A crosses the upper and lower boundaries of the segment. Furthermore, the implicit requirement that the lines be orthogonal *to each other* may impose certain limits of the types of solutions possible. Therefore, while the problem is somewhat simplified by the requirement that the lines be orthogonal to the axes, it does not appear to obviously reduce the complexity of the problem. This special case of the problem will not be dealt with further in this report, as it falls outside of the scope of the research dealt with, but it clearly deserves further investigation.

4.7 Choice

Despite the restrictions placed upon the forms of both allowable adjacencies and potential axial line placements in deformed urban grid configurations, it is still possible to find choice within a set of axial lines in a deformed urban grid. This is primarily because of the discrepancy between (half-) diagonal and thoroughfare lines; there is strong potential for both lines to cover mostly similar adjacencies with some differences, and these differences allow for choice. Consider Figure 4.7.

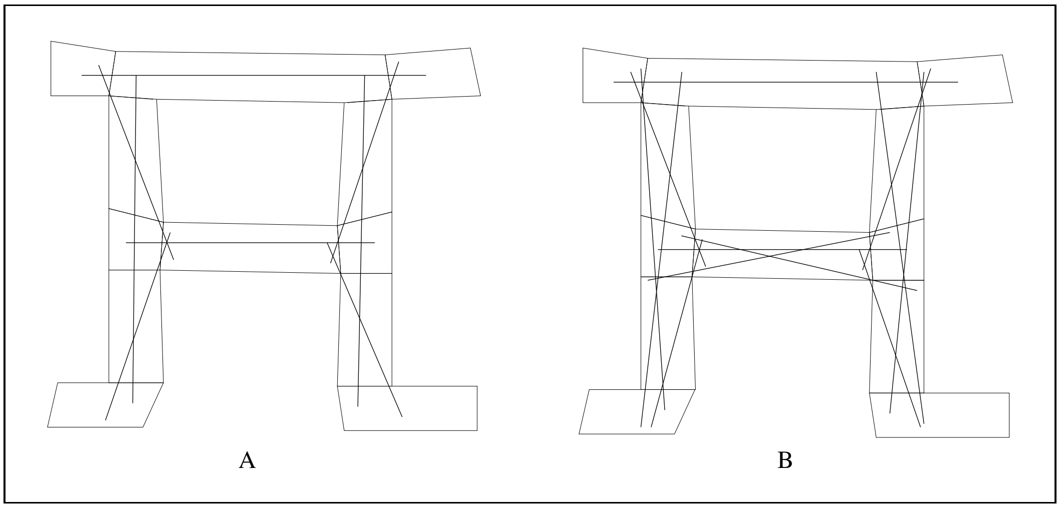

Figure 4.7: Choice Lines in a Deformed Urban Grid Configuration

In the configuration marked A in Figure 4.7, a simple deformed urban grid case is shown to have a degree of choice. In this example, either the four diagonal lines or the four thoroughfare lines cover all adjacencies in the grid. Alternatively, the two vertical thoroughfare lines and the two top diagonal lines could cover the grid. However, if the top thoroughfare were to be removed, leaving only its left and right intersections in place, only three of the thoroughfare lines would be required to cover the configuration, while all four diagonal lines would be required. If the top thoroughfare structure was duplicated on the bottom, it would require five thoroughfare lines, or four of the diagonal lines. So, clearly, it does not suffice to simply choose either method of covering the configuration. Further, there seems to be no simple way to choose between the line sets. The configuration marked B in Figure 4.7 illustrates what the line sets are likely to look like in an actual application. Again, there seems to be several ways to cover this grid, but no clear way to choose between them – a mix of diagonal and thoroughfare lines is also possible.

In addition, it is possible to "stack" the structure shown in Figure 4.7. Figure 4.8 shows how two such structures can be connected so that they are independent of each other. This is because the adjacency marked A cannot be cut by an axial line in either of the two substructures; hence, an axial line that cuts it must be added, and this axial line is easily extended so that it crosses all the other adjacencies in the "join" part of the configuration, effectively removing them from the problem. This demonstrates clearly that there could be a number of independent choice problems within the axial line placement problem such that each of them must be solved individually. Although only one particular choice configuration was used twice in Figure 4.8, it is not inconceivable that there are many – perhaps infinitely many – such configurations, and there seems to be no reason why they could not be stacked in a similar way. In particular, while Figure 4.8 stacks two identical choice configurations, either one could equally validly be one of the alternate forms of that configuration (as already discussed), with their corresponding differing solutions.

An even more problematic observation is that the impact of local choice does not necessarily have to stay local. While the choice factor in Figure 4.8 does not traverse choice units – meaning that the local choice made in one part does not affect the other – this is not necessarily the case. The possibility that

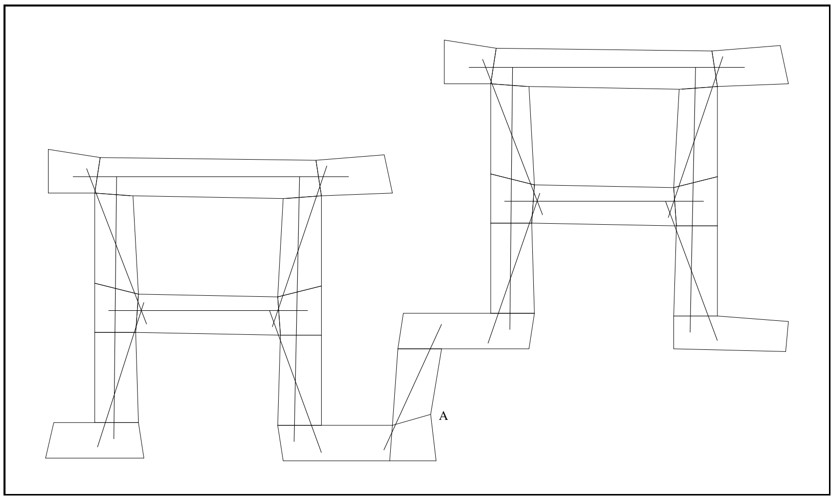

Figure 4.8: Stacking Choice Configurations

the choice made in one part of the grid may directly affect the choices made in other, seemingly separate parts of the grid shows clearly that even if a single choice unit can be resolved in polynomial time the ramifications of that choice could lead to a globally suboptimal solution, leading to the need to examine the ramifications of all combinations of all choice units, which is clearly not achievable in polynomial time. Figure 4.9 is an example of a way that the choice unit already given could be combined to form a single, larger, choice unit.

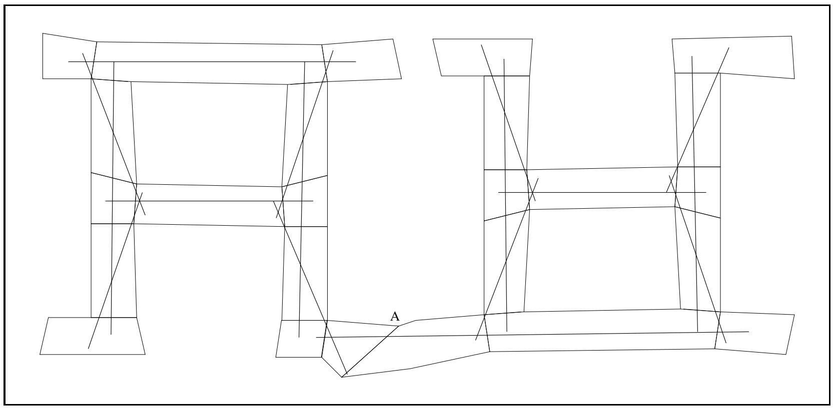

Figure 4.9: Combining Choice Units to Form Larger Choice Units

The adjacency labelled A in Figure 4.9 is required to be crossed by some axial line. If the first choice unit is considered separately, and solved without including the bottom left diagonal line, then the second choice unit is required to include its lowest horizontal thoroughfare line. Therefore, the choice in the first unit clearly can determine the choice made in the second, effectively rendering the compound unit into one, larger, choice unit. While this is a simple of example of such a compound choice unit, it can be made arbitrarily long by "chaining" choice units to either side in a similar fashion to the two in Figure 4.9. Such a choice unit could span the entire grid, where a choice made locally in the first original unit could force a possibly non-optimal choice in a unit some length down the chain. This situation clearly shows that there are situations in which the choice involved is neither simple nor local.

Since a grid could be densely covered, it would not be feasible to pick a line, trying it as included or not included, and comparing the results, since in the worst case there could be n lines crossing each adjacency, where n is proportional to the length or height of the grid. Removing a particular line cannot therefore be guaranteed to break the deadlock by "cascading" – causing the lines around it to be trivially discardable or essential, and this effect rippling throughout the grid. Each attempt that tests the effects of a line being off and on doubles the computation, and if this happens anything more than a constant number of times the complexity of the computation will be easily exponential.

Finally, it is important to note that although the compound choice units shown here are connected in relatively simple ways, there is nothing to prevent them from being connected in more complex ways. Each axial line can be of length proportional to the number of polygons in the grid in the worst case, and each adjacency it crosses could be crossed by several other axial lines, each of which are either a direct part of a choice unit or affect some other distant choice unit. Therefore a single axial line's presence or absence may affect the potential optimality of a solution arbitrarily far away in the grid.

It is clear from this analysis that there is no simple way to resolve the choice that can occur in placing axial lines in such configurations. This is especially significant in the light of the original conjecture that the axial line placement problem is solvable in polynomial time, which argued that the choice involved would be limited by the types of lines allowed in such a configuration [Sanders 2000]. An algorithm that resolves the choice depicted here would require significantly more investigation into the problem and significant insight into it that is outside the scope of this research. Therefore, within the context of the time allowances and scope of this research, the research hypothesis must be rejected.

4.8 Conclusion

This chapter has set out the crucial properties of both the partitioning adjacencies allowable in deformed urban grid configurations, and the forms of axial lines that can be placed within them. It has defined the necessary terminology and provided a basis for later methodical investigation into this significant problem area.

More importantly, it has shown that despite the conjecture that the restrictions imposed by deformed urban grid configurations would lead to a polynomial time solution, the deformation can create situations which contain choice that cannot be easily resolved. Because of this, it is likely that the problem cannot be solved in polynomial time. Even if it can, it is unrealistic to attempt to find a polynomial time algorithm for finding exact solutions to the axial line placement problem in deformed urban grids within the time bounds of this research. The demonstration of this choice therefore leads to the rejection of the research hypothesis, within the context of this research.

The next section, therefore, makes use of the insight gained into the problem in this chapter to present a heuristic algorithm for placing axial lines in deformed urban grids that produces approximate solutions in polynomial time.

Chapter 5

A Heuristic Algorithm

5.1 Introduction

This chapter presents a heuristic algorithm that produces an approximate solution to the axial line placement problem in deformed urban grids in $O(n^4)$ time. The algorithm presented aims to be both relatively efficient and relatively accurate, and may have significant practical application since there does not seem to be a clear way to solve the problem exactly in polynomial time. The algorithm is, however, structured in terms of an algorithm framework that can be used as the basis for an algorithm that does solve the problem in polynomial time, should further research reveal that the problem is, despite the difficult choice resolution problems posed, solvable in polynomial time. This framework reduces the problem to a case of removing the deadlock inherent in resolving choice. Further research would then only be required to focus on solving a simpler problem with at least some existing work to base further insights on.

Section 5.2 presents the algorithm framework that will be used to derive a heuristic algorithm, and could be used as the basis for a polynomial time exact solution to the axial line placement problem for deformed urban grids. Section 5.3 then presents the heuristic algorithm itself, by first presenting the heuristics used and decisions made, and then presenting the pseudo-code for each phase of the algorithm. Section 5.4 determines the time complexity of the algorithm, and is followed by section 5.5, which characterises the likely properties of the solutions generated by the algorithm. Finally, section 5.6 summarises.

5.2 A Framework

An algorithm to solve the axial line placement problem in deformed urban grids could consist of the following three stages:

1. **Line Placement** – all axial lines allowable within the configuration (and therefore potentially useful) are maximised and placed in a candidate set. This is done on a thoroughfare basis, since we know that all of the lines be aligned along a thoroughfare with the possible addition of a bridge on either end. There would be four stages, for each type of potential line placement, although these stages could easily be combined to increase the efficiency of the algorithm. Although this is a somewhat wasteful approach (lines are placed that will later be removed) it provides a useful framework for dealing with the problem.

2. **Line Filtering** – all essential lines are moved to an separate set, and their adjacencies marked as owned, and all redundant lines are removed from the candidate set, using a set of "line elimination rules". This process continues until a "deadlock" is reached where no further lines can be immediately marked as redundant or essential. The line elimination rules are:

(a) Any line that is the only line to cross an adjacency should be removed from the candidate set and added to the essential set.

(b) Any line that is a subset of another line can be removed from the candidate set.

(c) Any line that is an "unowned subset" of another line (all of its unowned adjacencies are crossed by a single other line) can be removed from the candidate set.

(d) Any line that crosses no unowned adjacencies can be removed from the candidate set.

(e) If two lines A and B cover (cross all the unowned adjacencies of) a third line C, and A and B are the only two lines to cross adjacency a, then C can be dropped from the candidate set.

3. **Deadlock Removal** – one line out of the deadlocked set is removed, in an attempt to resolve the choice inherent in a deadlock. The algorithm would then revert to the line filtering stage, continuing on in order to completely carry through the consequences of the choice made here, before reverting to this stage again. This part is where a polynomial time solution is unlikely to exist.

This approach was chosen because it was the best way to provide insight into the problem. The first stage handles all of the visibility aspects of the problem, reducing the problem to a somewhat more abstract form, and clarifying the approach used for the remainder of the algorithm. However, in the actual implementation of the algorithm, some parts of the first and second, and second and third phase may be merged in order to achieve an order reduction. For example, it may be possible to place all of the lines such that none of them can be thrown out using rule 2b.

Since it appears that no polynomial time algorithm to achieve the third phase of the algorithm is possible (see chapter 4), some parts of the algorithm will need to be replaced by efficient polynomial time approximations.

5.3 An Efficient Heuristic Algorithm

In order to produce an algorithm that can run in polynomial time, a heuristic is required to resolve the deadlock encountered in phase three, and certain parts of phases one and two are required to be removed. Therefore, the following heuristics were chosen as part of an efficient heuristic algorithm:

1. Only the longest diagonal line in each direction from each intersection will be included in the candidate set. This is because there are $O(m^2)$ possible diagonal lines for each thoroughfare of length m, which makes the size of the candidate set of a grid of size n potentially $O(n^2)$, and that makes the implementation of the line elimination rules expensive.

2. Rule 2e of the line elimination rules will be removed, since it is expensive to test and is a relatively obscure case.

3. Rule 2b can be ignored, since any line that satisfies it must also obviously satisfy rule 2c.

4. When a deadlock situation is reached (phase three), the line that crosses the fewest number of unowned adjacencies will be removed.

The first two changes to the heuristic algorithm seem to make sense when considered in the light of the deadlock resolution heuristic chosen. The possible consequences of these omissions will be examined later, in section 5.5.

5.3.1 Preliminaries and Data Structures

The algorithm will work within a structure of a set of horizontal and vertical thoroughfares, each having an array of adjacencies. All adjacencies will be given a unique number, ranging from 1 to n, where n is the number of adjacencies. This implies that a single number may appear in more than one thoroughfare structure, since a single adjacency may form part of more than one thoroughfare. We will label the thoroughfare structures v_i, where i = $1...m_v$, for vertical thoroughfares, and h_i, where i = $1...m_h$, for horizontal thoroughfares. Each adjacency will be represented by v_{ij} (or h_{ij}), where j = $1...k_{v_i}$ (or k_{h_i}). In addition, each vertical thoroughfare v_i has a corresponding list of associated left and right bridges, L_{if} ($f = 1...l_{v_i}$) and R_{ig} ($g = 1...r_{v_i}$), respectively. A similar structure exists for horizontal thoroughfares.

In addition, the algorithm will make use of the following three primary data structures to make the computation efficient:

1. `E` = $\{e_i\}$, the set of essential lines. This is a linked list of line structures, each of which contains a boolean array e_i.`cuts`, where e_i.`cuts[j]` is `true` if e_i cuts line j, or `false` otherwise. A pointer is kept to the end of this list so that additions can be made in constant time.

2. `C` = $\{c_i\}$, the set of candidate lines. This is also a linked list of line structures, but where each contains an integer array, c_i.`cuts`, where c_i.`cuts[j]` is 1 if c_i cuts unowned line j, -1 if it cuts line j but j is already owned, and 0 if it does not cut line j. Each element in `C` also contains an integer, `unowned-length`, which gives the unowned length of the line. A pointer is also kept to the end of this list to facilitate easy insertion.

3. `A`, an array of integers such that `A[j]` holds the number of times line j is crossed, or -1 if line j is owned.

The algorithm given in Figure 5.1 adds a candidate line to the set `C`. Note that although additional work needs to be done in order to maintain the list of adjacencies the line crosses, this is essential in order to recover a solution from the set of lines left in `E` when the algorithm is finished.

```
  Given:  F = {f_i}, i = 1...G, the lines this adjacency crosses.

1 Allocate new candidate line structure, K

2 for i = 1 to n
3   K.cuts[i] = 0;
4 end for
5 K.unowned-length = G;

6 for i = 1 to G
7   A[f_i] = A[f_i] + 1;
8   K.cuts[f_i] = 1;
9 end for

10 Add K to C.
```

Figure 5.1: Adding a Candidate Line

The algorithm creates a new axial line structure (line 1), initializes it (lines 2–5) and updates its `cuts` member and `A` in the loop in lines 6–9. Finally, the new structure is added to `C` (line 10). Note that the algorithm in Figure 5.1 assumes that no adjacency lines are as of yet owned. This is because it is called only in the first phase, and adjacencies are only marked as owned in the second phase, when axial lines are marked as essential.

Figure 5.2 gives the algorithm for marking a line as essential. A new essential line structure is created (line 1), and then a loop over lines 2–14 checks if each adjacency is cut by the line. If it is, then the new structure is updated to reflect that (line 4), and A is updated to mark that line as owned (line 5). In addition, all the other axial lines which may cross that line are updated – if they do, their `cuts` structure is updated and their `unowned-length` reduced (lines 6–11). Finally, the candidate line is removed from `C` and added to `E` (lines 15 and 16).

```
  Given:  Cand, the candidate line (in C) to be marked essential.

1 Allocate new essential line structure, K

2 for i = 1 to n
3   if Cand.cuts[i] ≠ 0 then
4     K.cuts[i] = true;
5     A[i] = -1;

6     for each cand ∈ C do
7       if cand.cuts[i] == 1 then
8         cand.cuts[i] = -1;
9         cand.unowned-length = cand.unowned-length - 1;
10      end if
11    end for

12  end if
13  else K.cuts[i] = false; end else
14 end for

15 Remove Cand from C, and free memory
16 Add K to E
```

Figure 5.2: Marking a Line as Essential

Finally, Figure 5.3 gives the algorithm for removing a line from the candidate set C, without marking it is essential. This algorithm is used when a line is “thrown away” by one of the line filtering rules. The loop in lines 1 to 5 simply decrements the count in the `A` structure if the line has crossed there, and the candidate line is removed from `C` in line 6.

The algorithms given so far are sufficient for all the necessary data structure manipulation in the remainder of the algorithm. However, the data structures `C`, `E` and `A` have to be initialised. Therefore, before the algorithm starts, `C` and `E` are set to empty, and `A[j]` is set to 0, for `j` = $1...n$. The first phase (line placement) then proceeds to add all of the possible potential lines to the candidate set and initialise `A` so that it holds the total number of lines crossing each adjacency.

5.3.2 Line Placement (Phase One)

The line placement phase is handled by two main algorithms (one vertical and one horizontal), each of which has four sub-algorithms. Therefore, we will handle only the vertical case, and leave out similar sub-algorithms. The algorithm given in Figure 5.4 presents line placement algorithm for the vertical case

```
   Given:  Cand, the candidate line (in C) to be deleted.

for i = 1 to n do
  if cand.cuts[i] == 1 then
    A[i] = A[i] - 1;
  end if
end for

Delete cand from C.
```

Figure 5.3: Deleting a Candidate Line

(the horizontal case is similar), giving pseudo-code for the `AddThoroughfareLines` and `AddDiagonalLines` procedures. The rest are omitted, as they are similar.

The first six lines simply loop through the vertical thoroughfares, and execute the line adding procedures on each. Lines 7–15 make up the `AddThoroughfareLines` procedure; it loops through each adjacency, constructs the visibility polygon from that adjacency to the end of the thoroughfare, and then loops to find the last adjacency in that visibility polygon. The procedure is then repeated in the other direction.

The `AddDiagonalLines` procedure (lines 16–25) is similar; however, it loops through the left bridges and constructs the visibility polygon from each one to the farthest right bridge, including the other bridges. It then loops to find the furthest of these within the visibility polygon. This is also repeated in the other direction.

5.3.3 Line Filtering (Phase Two)

Figure 5.5 presents the pseudo-code to perform phase two of the algorithm, and apply rules 2a, 2c and 2d as given in section 5.2. The line filtering phase is implemented as a single loop in order to clarify the code. This outer loop (lines 1 – 28) halts when no lines are removed or made essential – at that point, the deadlock resolution section of the algorithm should be applied. The loop in lines 3–13 implements rule 2a – it searches for an adjacency with only one crossing line, locates the line, and marks it as essential. The second loop (lines 14–20) implements rule 2d, and deletes those lines that cross no unowned adjacencies, and therefore have an `unowned-length` of zero. Finally, the loop in lines 21–27 implements rule 2c, which throws away lines that are unowned subsets of other lines.

5.3.4 Deadlock Removal (Phase Three)

The deadlock removal phase can be accomplished with a simple linear search that scans for the line with the smallest `unowned-length`. This line is then thrown away using the candidate removal algorithm given in Figure 5.3. After each deletion, the algorithm should then move back to the Line Filtering Phase, given in Figure 5.5.

5.4 Complexity

The complexity of each of the phases of the algorithm are best analysed individually.

```
for i = 1 to m_v do
  AddThoroughfareLines(i);
  AddHalfDiagonalLines(i);
  AddDiagonalLines(i);
  AddSideLines(i);
end for


Procedure AddThoroughfareLines(i)
for j = 1 to k_{v_i} do
  Construct the visibility polygon P from v_{ij} to v_{ik_{v_i}}
  t = k_{v_i};
  while v_{it} not in P do t = t - 1; end while;
  Add line from v_{ij} to v_{it} to C (see Figure 5.1)
end for
Repeat procedure backwards.
End Procedure

Procedure AddDiagonalLines(i)
for j = 1 to l_{v_i} do
  Construct visibility polygon P from L_{ij} to R_{ir_{v_i}} (including bridges)
  t = r_{v_i};
  while R_{it} not in P do t = t - 1; end for
  Add the line from L_{ij} to R_{it} to C (see Figure 5.1)
end for
Repeat procedure backwards.
End procedure
```

Figure 5.4: Line Placement (Vertical)

5.4.1 Preliminaries and Data Structures

The initialisation of C and E to the empty set can be done in constant time, while the initialisation of all elements of A to zero can be done in $O(n)$ time. In the algorithm to add a candidate line, given in Figure 5.1, line 1 takes constant time, as does line 10, since we are using a linked list. The loop in lines 2–4 executes exactly n times, and the loop in lines 6–9 executes G times. Since G is $O(n)$, the complexity of the entire operation is then $O(n)$.

In the algorithm that marks lines as essential (see Figure 5.2), the lines 1, 15 and 16 again take constant time, and the outer loop (lines 2–14) executes n times, and each `if` statement potentially running a loop (lines 6–11) that takes time in $O(|C|)$, which is the size of the candidate set. Since $|C|$ is $O(n)$, the entire operation takes time in $O(n \times n) = O(n^2)$. The algorithm to delete a candidate line (Figure 5.3) is a simple loop over n, and therefore takes time in $O(n)$.

5.4.2 Insertion Phase

The insertion phase has an outer loop that runs m_v times in the vertical case and m_h times in the horizontal case. This calls four sub-procedures. The `AddThoroughfareLines` procedure executes k_{v_i} times, and each execution constructs the visibility polygon for a chain of convex polygons, which can be done in $O(n)$ [Soltész 2000], and runs a linear loop, the last execution of which adds a line to the

```
do
  changes = false

  for i = 1 to n do
    if A[i] == 1 then
      for each cand ∈ C do
        if cand.cuts[i] then
          Mark cand as essential (see Figure 5.2)
          changes = true
          Restart outer loop
        end if
      end for
    end if
  end for

  for each cand ∈ C
    if cand.unowned-length == 0 then
      Delete cand (see Figure 5.3)
      changes = true;
      Restart outer loop
    end if
  end for

  for each cand1, cand2 ∈ C, cand1 ≠ cand2 do
    if cand1 is an unowned subset of cand2 then
      Delete cand1 (see Figure 5.3)
      changes = true;
      Restart outer loop
    end if
  end for

while changes == true
```

Figure 5.5: Line Filtering

candidate set, which is $O(n)$. The test within the loop can be done in constant time, since we are checking an edge for inclusion in the visibility polygon between two points that are already known. The inner body of the loop is therefore $O(n + k_{v_i} + n) = O(n)$, and the entire procedure therefore $O(k_{v_i} \times n)$. A similar argument find that the `AddDiagonalLines` complexity is $O((l_{v_i} + r_{v_i}) \times n)$. The `AddHalfDiagonalLines` is of complexity $O((l_{v_i} + r_{v_i}) \times n)$, and `AddSideLines` is of complexity $O(2(l_{v_i} + r_{v_i}) \times n)$.

In order to obtain the final complexity of the line insertion phase, it is necessary to notice that $\Sigma_{i=1}^{m_v} k_{v_i}$ is $O(n)$, since the total number of adjacencies in all of the vertical thoroughfares cannot exceed the total number of adjacencies in the entire grid. A similar argument holds for l_{v_i} and r_{v_i}, and for m_h. The total complexity of the insertion phase is the sum to m_v (and m_h) of the complexities of the sub-procedures; therefore, the total complexity of the insertion phase is $O(n^2)$. In addition, it is important to note that each adjacency adds only one axial line, so the size of `C` is going to be proportional n.

5.4.3 Line Filtering

The loop from lines 3–13 executes n times, with each execution potentially running an `if` statement that runs in $O(|C|) = O(n)$. The final `if` statement may also mark a line as essential (which is $O(n^2)$), so the complexity of this loop is $O(n \times n + n^2) = O(n^2)$.

The loop from lines 14–20 executes $O(n)$ times, with the final execution possibly executing a deletion, which is $O(n)$. Therefore the total complexity of this loop is $O(n + n) = O(n)$.

The loop from lines 21–27 executes $O(n^2)$ times, in order to obtain the pair `cand1` and `cand2`. Each execution calls a subroutine that checks whether one line is an unowned subset of another, which is $O(n)$ (line 22). In addition, the final check may execute a line deletion, which is $O(n)$. Therefore, the total complexity of the loop is $O(n^2 \times n + n) = O(n^3)$.

The outer loop (lines 1–28) may execute $O(n)$ times; therefore, the complexity of the line filtering process is $O(n \times (n^2 + n + n^3)) = O(n^4)$.

5.4.4 Deadlock Resolution

A loop to find the line with the smallest value of `unowned-length` takes time in $O(|C|)$, while the deletion of that line takes time in $O(n)$, so the removal of the line takes time in $O(n + |C|) = O(n)$ since $|C|$ is $O(n)$. This may happen $|C|$ times, once for each candidate line (except perhaps a small constant number of them) so the total complexity for this phase is $O(n^2)$.

5.4.5 Overall Complexity

The overall complexity of the algorithm is $O(n + n^2 + n^4 + n^2) = O(n^4)$, and is clearly dominated by the complexity of the line filtering process. Only trivial cases would allow for the finding of a solution without encountering the third rule of the line filtering algorithm; those cases could have no deadlock, and be entirely resolvable with the use of essential line selection and dead line removal. Since this case is, however, a possibility, we will consider the best case of the algorithm to be $O(n^3)$, and the worst and average cases $O(n^4)$. In reality, the best case could be better (a constant number of lines covering the grid), but since such cases are pathological we will not consider them further.

5.5 Expected Solution Characteristics

The algorithm given in this chapter should produce better solutions than the previous algorithm for this problem [Soltész 2000], since the previous algorithm can be considered the same as the one given, with the line filtering rules, deadlock removal, diagonal, half-diagonal and side-line placement sections removed. The previous algorithm requires no line filtering or deadlock removal since it includes only a very small subset of the allowable axial lines (thoroughfare lines only), in which no deadlock or extraneous lines can occur. However, the previous algorithm is an order of n faster [Soltész 2000], and therefore could potentially be useful in situations where accuracy is not the primary concern.

Of the four heuristics used (see section 5.3), only three of them can impact the quality of the potential solutions produced by the algorithm. The first, which includes only the longest diagonal lines, can only impact the solution when both ends of a diagonal line that is not the longest one from either of its bridges cause the line to be marked as essential. This is because diagonal lines must exist from both bridges that are longer than this line, in the same direction, and therefore cover all of its adjacencies except those contained in the bridges at the other end. If only one of these makes the line essential, it can be replaced by one of the longer diagonal lines without affecting the solution. The situation where both ends of the diagonal line make it essential is unlikely to occur in practice, especially since we use a heuristic which favours longer lines.

The heuristic which removes the line with the smallest unowned length makes sense because such a line has the fewest opportunities to become essential, as each of an axial line's unowned adjacencies represent a chance for that line to become essential. Should this heuristic fail, it means that more than one of the adjacencies that the line crossed would have made it essential; since otherwise, it could only have added one other line to the solution, and not affected its optimality. The line with the fewest unowned adjacencies seems to offer the least likelihood of this. However, an empirical investigation into the quality of the solutions generated by this algorithm could provide more insight into the nature of the solutions generated by it.

5.6 Summary

This chapter has presented a heuristic algorithm that can be used to solve the problem of axial line placement in deformed urban grids. It is based on an algorithm framework that could be useful for further work on the problem, and could potentially be used as a basis for an exact solution. The algorithm given has been presented in phases, with an overall complexity of $O(n^4)$, where n is the number of adjacencies in the grid, and some of the cases in which it is likely produce bad approximations have been outlined.

Chapter 6

Discussion

6.1 Introduction

Although the research presented in this report indicates that the issue of resolving choice in axial line placement in deformed urban grids appears to make the problem NP-Complete (at least within the scope of this report), it does not constitute a definitive answer to the question of whether or not the problem can be solved in polynomial time. One of the major results of the presented research is a body of work that could provide the basis for more in-depth work that either proves that the problem is NP-Complete or finds a polynomial-time algorithm for resolving choice. This primary focus of this chapter is to discuss the implications of this research for the problem and further research on it.

The next section briefly outlines the significance of the results presented, both in terms of the aims of this research and in terms of future work on the problem. Section 6.3 briefly outlines the limitations of this research, and the areas in which it might be improved upon. Section 6.4 details what would be required from future work that attempted to modify the framework algorithm presented in chapter 5 in order to obtain an exact solution, while section 6.5 lists some of the NP-Complete problems that future research might attempt to transform the problem to in order to prove that it is also NP-Complete. Finally, section 6.6 concludes.

6.2 Significance of Results

The results presented are significant in that they provide a basis for further research into the axial line placement problem, by developing some of the terminology necessary to deal with the problem, proving some of the constraints that were thought to exist in the problem were indeed there, and by illustrating that despite these, the resolution of choice is still the crucial factor that makes the problem difficult. In addition, the provision of a heuristic algorithm provides a useful practical tool that could potentially contribute towards the automation of the space syntax method.

6.3 Limitations

Unfortunately, due to the nature of both the scope and time constraints placed upon this research, it does not constitute a definitive answer to the question of whether or not the problem of axial line placement in deformed urban grids is actually solvable in polynomial time. While this research shows that the issue of choice remains a dominant one, no in-depth evaluation of the forms of choice or attempts at resolving it could be performed within the time allocated for this report. In addition, the level of knowledge required for an NP-Completeness proof by transformation was also probably outside of the scope of this research. Hopefully, however, either problem could be within the scope of a future Honours research report, using the results presented here as a starting point.

6.4 Requirements for an Exact Solution

In order for an exact solution to be obtainable in polynomial time, some method must be discovered that resolves the choice inherent in the problem (and outlined in section 4.7) in polynomial time. The framework algorithm given in chapter 5 provides a good starting point for this, since by the start of the third phase (deadlock resolution) it has abstracted the problem from a geometric problem to a problem of minimum set cover, and removed all of the easily characterisable cases. In particular, the algorithm for deadlock resolution is likely to be one of the following types:

1. Selecting a line to remove from the candidate set – where it can be shown that there is some polynomial-time way of determining which of the set of lines remaining in the candidate set should be dropped. A proof would require demonstrating that either the line cannot be in the solution set, or its removal would include only one other line in the solution set, and it therefore does not affect the optimality of the solution.

2. Selecting a line to mark as essential – where it can be shown that there is some polynomial-time way of finding a line in the candidate set that is guaranteed to either be in the solution set, or to remove a line from the solution set if added.

3. Finding a direct deadlock resolution algorithm – taking advantages of the nature of the deadlocks likely to occur within deformed urban grid configurations, an algorithm would be produced that could be shown to find a way to break the deadlock while guaranteeing an exact solution. One approach might be to examine the ring-like structure of deadlocks in deformed urban grids, and investigate ways to directly find a minimal set of lines in the ring that covers all of its adjacencies. Another approach might be to set each line as "off" and "on", and comparing the solutions generated thereby. However, this method would be required to show that only a constant number of lines would have to be "switched" in order to find a solution for each deadlocked set, because otherwise the complexity of such a function becomes $\Theta(2^n)$.

While none of these approaches offers an immediately obvious solution method, each of them opens up a potentially viable research path; however, another research path more directly suggested by the results of this research would be attempting to actually prove that the problem is NP-Complete, by transforming another NP-Complete problem to it.

6.5 NP-Completeness Transformations

An NP-Completeness proof would be the logical continuation of this research, since it would show that a polynomial time algorithm is extremely unlikely to exist. Such a proof would require a transformation from an arbitrary instance of an NP-Complete problem to an instance of the axial line placement problem for deformed urban grids, that can be shown to take polynomial time. Therefore, any attempt to prove the problem NP-Complete would require the selection of a problem that has already been shown to be NP-Complete and seems to be suitable for the transformation. There are two likely candidates for this task.

Vertex Cover was transformed to both instances of the axial line placement problem that have been shown to be NP-Complete [Sanders *et al.* 1997; Sanders 1999]. In these cases, the special form of biplanar vertex cover was used. It therefore seems natural that if the deformed urban grid case is to be shown to be NP-Complete, vertex cover would be a good problem to transform to it.

Minimum Cover also seems to be a natural fit, as has already been discussed in chapter 2. The particular case that seems to merit a strong examination is that of minimum cover when all sets have three or fewer elements in them. It would seem that a case could be constructed where each of the unique elements in the subsets could be single adjacency lines, and each of the subsets represented by lines going through them. The necessary extra adjacencies required to make the configuration legal could be added without affecting the solution by crossing them with artificial axial lines that could be made essential – and therefore can be discarded from the final solution while keeping the minimum cover solution intact. However, it seems unlikely that such a simplistically direct transformation would work – it would be difficult to directly associate a set with a single axial line since the "connectivity" between sets might be more than is allowable in a deformed urban grid configuration. Therefore, a set of axial lines could be used, only one of which would be required to find a solution. The presence of any of those lines in the actual solution could mean the inclusion of a subset in the minimum covering set. Clearly, an NP-Completeness proof of this form would require a relatively sophisticated transformation procedure and proof. Although this was outside of the scope of this research, the properties of deformed urban grids presented in this report could form the basis for it.

6.6 Conclusion

Although this research has not shown conclusively that the axial line placement problem in deformed urban grids has no polynomial time solution, it has provided significant evidence that the conjecture that the form of the problem provided a simple way to resolve the issue of choice is not correct. Furthermore, by developing the properties of deformed urban grids, identifying the issues of choice that remain, and presenting as framework algorithm for further development, it has provided a significant basis for further work and proposed starting points for such work.

Chapter 7

Conclusion

7.1 Introduction

This chapter concludes this research report by presenting an brief overview of the main research results detailed in the document so far. It examines the significance of the outcomes of the research in terms of both providing a basis for future research and the potential practical ramifications of the research. An outline of the results of the research is also given, in terms of the research hypothesis, and areas in which future work may prove fruitful are discussed.

Section 7.2 relates the results presented to the hypothesis, while section 7.3 outlines the significance of those results. Section 7.4 presents an outline of potential future work, and 7.5 concludes this report.

7.2 Results

The primary result of this research was that the axial line placement problem in deformed urban grids is significantly harder than has been recently conjectured, leading to the conclusion that it is probably NP-Complete. This result stems from the demonstration of the presence of a significant amount of choice despite the restrictions that the deformed urban grid class places on the axial lines that can be placed within it. Unfortunately, actually proving that the problem is NP-Complete is outside the scope of an Honours research report. Because of this, the research hypothesis (that the problem could be solved in polynomial time) was rejected, *within the context of this research*.

However, this research has developed the properties of deformed urban grids to a reasonable level, and therefore provided a reasonable basis for future work on the problem. The provision of a relatively well developed set of properties for the problem should significantly aid any further attempts to either prove the problem NP-Complete, or provide a polynomial time algorithm to solve it. In addition, a heuristic algorithm that has potential practical application was presented, and the characteristics of the solutions it derives were examined briefly. Therefore, in addition to providing a basis for further research, the research presented in this report has produced a practically applicable algorithm that may be useful in an actual implementation of the space syntax method.

7.3 Significance

The presence of choice in the axial line placement problem for deformed urban grids is a significant finding, because it demonstrates that even though the deformed urban grid configuration imposes significant constraints upon the forms of the axial lines that can be placed with it, it is still not likely to be solvable in polynomial time because it contains choice. Since this class of configurations has potential application, the research presented suggests the need for algorithms that generate approximate solutions for it in polynomial time. One of these has been presented in this document, and its characteristics briefly examined.

Even if further work is able to demonstrate that the form of choice present in deformed urban grid configurations can be resolved in polynomial time, the research presented above should at the very least be able to provide a basis for such research.

7.4 Future Work

The natural ideal for future work would be the discovery of an NP-Completeness proof that conclusively settles the question of whether or not the problem can be solved in polynomial time. Failing that, any work which provides a better characterisation of the choice inherent in the problem would contribute towards both the quality of later heuristic algorithms and further attempts to find a proof. Alternatively, if the problem turns out not to be NP-Complete, a polynomial time algorithm with a proof of correctness would also settle the issue.

However, there are other, less daunting, areas in which future work is potentially fruitful. The primary one would be in the development of better heuristic algorithms, and the empirical investigation of the performance of the algorithms presented in this research and in Soltész [2000]. In addition, since the partitioning problem for deformed urban grids is likely to be NP-Hard [Sanders 2000], a heuristic algorithm that exploited the specific characteristics of a particular heuristic partitioning algorithm (for example, Bukovska [2000]) could prove valuable for future implementations of the space syntax method. It is not unconceivable that an algorithm could be found that produced an exact solution for the problem on an inexact partition, even if the problem is shown to be NP-Complete in the general case. For example, if the adjacencies in the partitioning of a deformed urban grid are constrained to appear within intersection boundaries, the problem could become significantly easier.

Finally, there is some scope of the actual implementation of the space syntax method in the near future, and the empirical comparison of current heuristic algorithms for partitioning and axial line placement with the performance of human town planners.

7.5 Conclusion

The deformed urban grid case is clearly a special case worthy of attention: it has been shown to have significant application, and is the natural area for further research into axial line placement. The question of whether or not it can be solved in polynomial time is therefore an important research question that poses both a significant challenge and an interesting area of research. Through both the provision of a basis of further work, and the demonstration of the choice inherent in the problem, the work presented in this document has gone some way towards answering an important question that has relevance for the axial line placement problem in general, and for the eventual automation of the space syntax method.

References

[Adler *et al.* 2001] J. Adler, G.D. Christelis, J.A. Deneys, G.D. Konidaris, G. Lewis, A.G. Lipson, R.L. Phillips, D.K. Scott-Dawkins, D.A. Shell, B.V. Strydom, W.M. Trakman, and L.D. Van Gool. *Finding Adjacencies in Non-overlapping Polygons*. Electronic paper, *Proceedings of the SAICSIT Conference* – `http://osprey.unisa.wits.ac.za/saicsit2001`, 2001.

[Avis and Toussaint 1981] D. Avis and G. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Transactions on Computers*, 30(12):910–914, 1981.

[Baase and Van Gelder 2000] S. Baase and A. Van Gelder. *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley, Reading, Massachusetts, 3rd edition, 2000.

[Brassard and Bratley 1996] G. Brassard and P. Bratley. *Fundamentals of Algorithmics*. Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1996.

[Bukovska 2000] D. Bukovska. *Partitioning a Deformed Urban Grid*. Honours Research Report, Department of Computer Science, University of the Witwatersrand, November 2000.

[Du Plessis and Sanders 2000] N. Du Plessis and I.D. Sanders. *Partial Edge Visibility in Chains of Orthogonal Rectangles*. Technical Report TR-Wits-CS-2000-15, University of the Witwatersrand, 2000.

[El Gindy and Avis 1981] H. El Gindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *Journal of Algorithms*, 2:186–197, 1981.

[Garey and Johnson 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability: a guide to the theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, 1979.

[Hillier *et al.* 1983] B. Hillier, J. Hanson, J. Peponis, J. Hudson, and R. Burdett. Space syntax, a different urban perspective. *Architect's Journal*, (178):47–63, 1983.

[Hillier 1996] B. Hillier. *Space is the machine: A configurational theory of architecture*. Cambridge University Press, Cambridge, UK, 1996.

[O'Rourke 1987] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Number 3 in International Series of Monographs on Computer Science. Oxford University Press, New York, 1987.

[O'Rourke 1995] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, Cambridge, UK, 1995.

[O'Rourke 1997] J. O'Rourke. Visibility. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, Discrete Mathematics and Its Application, chapter 12, pages 467–479. CRC Press, Boca Raton, New York, 1997.

[Sanders *et al.* 1997] I.D. Sanders, D.J. Lubinsky, and M. Sears. Ray guarding configurations of adjacent rectangles. In *Proceedings of the 1997 National Research and Development Conference (SAICSIT 97)*, pages 221–238. Potchefstroom University of Higher Christian Education, 1997.

[Sanders *et al.* 2000] I.D. Sanders, D.C. Watts, and A.D. Hall. *Orthogonal Axial Line Placement in Chains and Trees of Orthogonal Rectangles*. Technical Report TR-Wits-CS-2000-25, University of the Witwatersrand, 2000.

[Sanders 1999] I.D. Sanders. Non-orthogonal ray guarding. *Abstracts for the Eleventh Canadian Conference on Computational Geometry*, pages 80–83, 1999. The full paper is available in the electronic proceedings – `http://www.cs.ubc.ca/conferences/CCCG/elec.proc/elecproc.html`.

[Sanders 2000] I.D. Sanders. Placing axial lines in urban grids. *The South African Computer Journal*, (26):145–153, November 2000.

[Sanders 2001a] I.D. Sanders, 2001. Personal Communication.

[Sanders 2001b] I.D. Sanders. *The Axial Line Placement Problem*. PhD thesis, University of Pretoria, 2001. Work in progress.

[Soltész 2000] É Soltész. *The Axial Line Placement Problem in Deformed Urban Grids*. Honours Research Report, Department of Computer Science, University of the Witwatersrand, November 2000.

Phelile.