

The axial line placement problem
by
Ian Douglas Sanders

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy
in the Faculty of Natural and Agricultural Science
University of Pretoria
Pretoria
June 2002

Supervisor: Professor D. G. Kourie, Department of Computer Science

In memory of my father,
Douglas Rutherford Sanders,
26 April 1927 – 18 October 1997

Acknowledgements

A number of people contributed in many different ways to the completion of this work.

- I would like to thank Prof. Glenn Mills (at that time of the Department of Architecture, University of the Witwatersrand) for bringing the problem to my attention and encouraging me to work on it.
- The work on this thesis started under the supervision of Prof. David Lubinsky and Prof. Michael Sears of the University of the Witwatersrand and continued under the supervision of Prof. Derrick Kourie of the University of Pretoria. I would like to thank them for the effort that they put into guiding this work.
- The School of Computer Science at the University of the Witwatersrand assisted me with paying my fees. This year I was also given a lighter workload in order to finish the thesis. I am very grateful for this.
- Some of the work was submitted to conferences and journals and the anonymous referees made comments which assisted me in improving my work. I would thus like to thank these people, particularly the South African Computer Journal referees, for their helpful comments.
- I would also like to thank Joseph O'Rourke for his kind email advice on the naming of my problem and Theresé Biedl for pointing out some related literature to me.
- Steve Hedetniemi and the "algorithms seminar group" at Clemson University gave me lots of encouragement during my sabbatical there in 1996. My thanks to all involved.
- Sigrid Ewert and Derrick Kourie assisted me with the translation of the summary into Afrikaans. I thank them for their efforts and accept full responsibility for any errors.
- I would like to thank my research students who tackled offshoots of the work. Their participation added a lot to my enjoyment of the work and provided stimulation for me at important times.



- My students past, present and future are the reason that I felt that need to start off on this path in the first place. I thus owe them a vote of thanks too.
- My family have always been supportive of my academic endeavours and warrant a vote of thanks for their support in this latest effort as well.
- I owe a very sincere vote of thanks to my friends, and colleagues, Scott Hazelhurst and Conrad Mueller who were always prepared to offer their comments on various pieces of the work and to offer me encouragement and support.
- Finally, my friends offered encouragement and support (typically “Get the bloody thing done!”) at various stages of the work. This applies particularly to Sheila Rock, Lex Holt (who also provided \LaTeX support), Vashti Galpin, Sean Pyott and András Salamon. Thank you all very much for keeping me on track and helping me get past the frequent “thesis avoidance” episodes.

Summary

Visibility, guarding and polygon decomposition are problems in the field of computational geometry which have roots in real world applications. These problems have been the focus of much research over a number of years. This thesis introduces a new problem in the field – The Axial line Placement Problem – which has some commonalities with these other problems. The problem arises from a consideration of the computational issues that result from attempting to automate the space syntax method. Space syntax is used for describing, quantifying and interpreting the spatial patterns in urban designs by analysing the relationship between the space through which one can move (roads, parks, etc.) and the buildings in the urban layout. In particular, this thesis considers the problem of the placing the axial lines, defining paths along which someone can move, to cross the shared boundaries between the convex polygons which represent the space through which someone can move in the town.

A number of simplifications of the original problem are considered in this thesis. The first of these is the problem of placing the smallest number of orthogonal line segments (orthogonal axial lines) to cross the shared boundaries (adjacencies) in a collection of adjacent orthogonal rectangles. This problem is shown to be NP-Complete by a transformation from the vertex cover problem for planar graphs. A heuristic algorithm which produces an approximation to the general solution is then presented. In addition, special cases of collections of orthogonal rectangles which allow polynomial time solutions are described and algorithms to solve some of these special cases are presented.

The problem where the axial lines, that pass through the adjacencies between orthogonal rectangles, can have arbitrary orientation is then considered. This problem is also shown to be NP-Complete and once again heuristic approaches to solving the problem are considered. The problem of placing axial lines to cross the adjacencies between adjacent convex polygons is a more general case of the problem of placing axial lines of arbitrary orientation in orthogonal rectangles. The NP-Completeness proof can be extended to this problem as well.

The final stage of the thesis considers real world urban layouts. Many urban layouts are regular grids of roads. Such layouts can be modelled as general urban grids and this thesis shows that it is possible to find the minimal axial line cover in

general urban grids in polynomial time. Some urban layouts are less regular and the idea of a deformed urban grid is introduced to model some of these layouts. A heuristic algorithm that finds a partition of a deformed urban grid in polynomial time is presented and it is conjectured that the axial map of a deformed urban grid can be found in polynomial time. The problem is still open for more general urban layouts which cannot be modelled by deformed urban grids.

The contribution of this thesis is that a number of new NP-Complete problems were identified and some new and interesting problems in the area of computational geometry have been introduced.

Opsomming

Sigbaarheid, waghou en veelhoek-dekomposisie is probleme in berekeningsmeetkunde wat hulle oorsprong in reële toepassings het. Die probleme is sedert jare die onderwerp van vele navorsing. Hierdie tesis voeg 'n nuwe probleem by die navorsingsgebied – die Asselyn Plasingsprobleem – wat sekere gemeenskaplikhede met bogenoemde probleme het. Laasgenoemde probleem vloei voort uit 'n beskouing van die berekeningskwessies wat ontstaan wanneer pogings aangewend word om die ruimte-sintaksis metode te outomatiseer. Ruimte-sintaksis word gebruik vir die beskrywing, kwantifisering en interpretasie van ruimtelike patrone in stedelike ontwerpe en wel deur die verwantskap tussen die ruimte waardeur 'n mens kan beweeg (paaie, parke, ens.) en die geboue in die stedelike uitleg te ontleed. Hierdie tesis beskou, in die besonder, die probleem van die plasing van asselyne op sodanig wyse dat hulle gedeelde grense tussen konvekse veelhoeke kruis, waarby the lyne paaie waarlang mens kan beweeg en die veelhoeke die ruimte waardeur mens deur die stad kan beweeg, verteenwoordig.

'n Aantal vereenvoudigings van die oorspronklike probleem word in hierdie tesis beskou. Die eerste hiervan is die probleem om die kleinste moontlike aantal ortogonale lynsegmente (ortogonale asselyne) op so 'n wyse te plaas dat hulle die gedeelde grense in 'n versameling van aangrensende ortogonale reghoeke kruis. Daar word gewys dat hierdie probleem NP-volledig is, deur 'n transformasie van die nodus-dekkingsprobleem (“vertex cover problem”) vir planêre (“planar”) grafieke na die problem uit te voer. 'n Heuristiese algoritme wat 'n benaderde oplossing tot die algemene probleem bied, word dan voorgestel. Addisioneel word spesiale gevalle van versamelings van ortogonale reghoeke wat polinomiese tyd oplossings toelaat beskryf. Algoritmes wat sekere van hierdie spesiale gevalle oplos word aangebied.

Daarna word die probleem beskou waarvolgens asselyne wat deur aangrensende ortogonale reghoeke gaan, arbitrêre orientasie mag hê. Hierdie probleem word ook as NP-volledig bewys en weereens word heuristieke benaderings om die probleem op te los, beskou. Die probleem om asselyne te plaas sodanig dat hulle grense tussen aangrensende konvekse veelhoeke te kruis is 'n veralgemening van die probleem om asselyne van arbitrêre orientasie in reghoeke te plaas. Die NP-volledigheidsbewys kan ook na die meer algemene probleem uitgebrei word.

Die finale fase van die tesis beskou die uitleg van reële stede. In die geval van baie stede is die uitleg 'n reëlmatige rooster van paaie. So 'n uitleg kan as 'n algemene stedelike rooster gemodeleer word en hierdie tesis toon aan dat dit moontlik is om die minimum asselyn dekking van sulke roosters in polinomiese tyd te bepaal. Sekere stede se uitleg is minder reëlmatig en die konsep van 'n verwronge stedelike rooster word voorgestel om sommige daarvan te modeleer. 'n Heuristiese algoritme wat in polinomiese tyd 'n partisie van 'n verwronge stedelike rooster vind, word aangebied. Daar word gepostuleer dat die assekaart van 'n verwronge stedelike rooster in polinomiese tyd gevind kan word. Die probleem vir stedelike uitlegte wat nie deur verwronge stedelike roosters gemodeleer kan word nie, bly egter steeds onopgelos.

Die bydrae van hierdie tesis is dat 'n aantal nuwe NP-volledige probleme geïdentifiseer is, en sommige nuwe en interessante probleme tot die gebied van berekeningsmeetkunde toegevoeg is.

Preface

Some of the work in this thesis has been previously published.

- The NP-Completeness proof in Chapter 4 was published in the South African Computer Journal [Sanders *et al.*, 1999].
- The axial line placement problem in chains and trees of orthogonal rectangles presented in Chapter 4 was also published in the South African Computer Journal [Sanders *et al.*, 2000b]. Much of the work for this paper was done under my supervision by two Computer Science Honours students, Claire Watts and Andrew Hall, as the research component of their degrees.
- The axial line placement problem in urban grids and deformed urban grids in Chapter 7 was accepted for the South African Institute of Computer Scientists and Information Technologists 2000 research symposium as a full research paper. It was published in a special issue of the South African Computer Journal [Sanders, 2000].
- The NP-Completeness proof in Chapter 5 was presented at the 11th Canadian Conference on Computational Geometry – an extended abstract was published in a collection of such abstracts and the full paper is available electronically [Sanders, 1999].
- The heuristics proposed in Chapter 5 were presented at the 13th Canadian Conference on Computational Geometry – an extended abstract was published in a collection of such abstracts [Sanders and Kenny, 2001a]. The full paper is available as a technical report in the School of Computer Science at the University of the Witwatersrand [Sanders and Kenny, 2001b]. Some of the work for this paper was done under my supervision by a Computer Science Honours student, Leigh-Ann Kenny, as the research component of her degree.
- Various presentations were made of “work in progress” at the Southern African Computer Lecturers’ Association annual conferences and the South African

Institute of Computer Scientists and Information Technologists annual research symposia [Sanders *et al.*, 1995, 1997; Sanders, 1998a,b; Bilbrough and Sanders, 1998].

- Some work has also been published as technical reports in the Department of Computer Science at the University of the Witwatersrand [Watts and Sanders, 1997; Sanders *et al.*, 2000a; du Plessis and Sanders, 2000; Sanders and Kenny, 2001b].
- My Honours students over the years have worked on some small parts of the overall research [Watts, 1997; Zarganakis, 1997; Soares, 1997; Wilson, 1997; Bilbrough, 1998; du Plessis, 1999; Hall, 1999; Ashman, 1999; Bukovska, 2000; Kenny, 2000; Soltész, 2000; Konidaris, 2001; Scott-Dawkins, 2001; Phillips, 2001; Hagger, 2001].

Contents

Acknowledgements	ii
Summary	iv
Opsomming	vi
Preface	viii
1 Introduction	1
1.1 Background to the problem	1
1.2 The scope for automation	3
1.3 The research focus of the thesis	10
1.4 Overview of the remainder of the thesis	10
2 Background	12
2.1 Introduction	12
2.2 Terminology	14
2.3 NP-Complete problems	24
2.3.1 Introduction	24
2.3.2 NP-Complete Problems	24
2.3.3 Proving the NP-Completeness of a new problem	26
2.3.4 NP-Hard problems	28
2.3.5 Summary	28
2.4 Relation of previous work to ALP	29
2.4.1 Overview	29
2.4.2 Short historical perspective	30
2.4.3 Putting <i>ALP</i> into context with other research	38
2.4.4 Results that informed the research on <i>ALP</i>	53
2.5 Conclusion	61
3 Research Questions	63
3.1 Possible Research Areas	63
3.2 Scope of this thesis	65

4	Placing orthogonal axial lines to cross adjacencies between orthogonal rectangles	68
4.1	Introduction	68
4.2	Statement of the Problem	69
4.3	Addressing the problem	69
4.4	Proving NP-Completeness of the problem of resolving choice	72
4.5	Heuristic Algorithm	84
4.5.1	Determining the adjacencies between the rectangles	86
4.5.2	Determining the axial lines	90
4.5.3	The Correctness of the method	96
4.6	Complexity Argument	97
4.6.1	Time	97
4.6.2	Space	100
4.6.3	Bounding the heuristic	100
4.6.4	Experimental Results	101
4.7	Special Cases that can be solved exactly in polynomial time	103
4.7.1	Mapping to interval graphs	103
4.7.2	Chains and trees of orthogonal rectangles	107
4.7.3	More general cases	126
4.8	Future research	126
4.9	Conclusion	127
5	Placing axial lines with arbitrary orientation to cross the adjacencies between orthogonal rectangles	130
5.1	Introduction	130
5.2	Statement of the Problem	130
5.3	Proving the problem is NP-Complete	132
5.4	Determining whether axial lines can be placed in chains of adjacent rectangles	139
5.5	Heuristics to find approximate solutions for <i>ALP-ALOR</i>	142
5.5.1	Overview	142
5.5.2	Extending lines into all neighbours	142
5.5.3	Separating top-bottom and left-right lines	144
5.5.4	Longest Chains	145
5.5.5	Crossing one adjacency at a time	149
5.5.6	Extending forwards and then backwards	156
5.5.7	Summing Up	159
5.6	Special cases which can be solved in polynomial time	159
5.7	Future Research	159
5.8	Conclusion	160

6	Placing axial lines with arbitrary orientation to cross the adjacencies between convex polygons	162
6.1	Introduction	162
6.2	Statement of the Problem	163
6.3	Proving the Problem is NP-Complete	163
6.4	Heuristics to find approximate solutions for <i>ALP-ALCP</i>	165
6.5	Special cases of <i>ALP-ALCP</i> which can be solved in polynomial time	165
6.6	Future Work	165
6.7	Finding the adjacencies in a configuration of adjacent convex polygons	165
6.8	Conclusion	167
7	Placing Axial Lines in Town Plans	168
7.1	Introduction	168
7.2	Urban Grids	169
7.3	Deformed Urban Grids	181
7.4	More general urban polygons	191
7.5	Conclusion	191
8	Future Research	194
8.1	Introduction	194
8.2	Open problems	194
8.3	Future research arising from this thesis	195
8.4	Conclusion	197
9	Conclusion	198
9.1	Introduction	198
9.2	Contributions of this thesis	199
9.3	Future work	200
9.4	Overall Conclusions	201

List of Figures

1.1	An example town plan	5
1.2	A zoomed view of a portion of the example town plan	6
1.3	A convex map of the enlarged version of the town plan with 24 convex spaces and its associated axial map with 6 axial lines	8
1.4	A convex map of the enlarged version of the town plan also with 24 convex spaces and its associated axial map with 7 axial lines	9
2.1	A simple polygon (Shermer [1992])	15
2.2	A star polygon – x is a kernel of the polygon	16
2.3	Comb polygons (Shermer [1992])	16
2.4	An orthogonally convex polygon and orthogonally convex star (Shermer [1992])	17
2.5	Orthogonal comb polygons (Shermer [1992])	18
2.6	A simple polygon and one of its triangulations (Shermer [1992])	18
2.7	A spiral polygon (Shermer [1992])	19
2.8	Point a can “see” b and c , but not d (Shermer [1992])	19
2.9	A pair of L_3 (or link-3) visible points (Shermer [1992])	20
2.10	The visibility polygon of the point y (shown as the darker shaded subpolygon) – y is the kernel of the visibility polygon but not of the original polygon	20
2.11	A covering guard set (Shermer [1992])	21
2.12	A hidden set (Shermer [1992])	22
2.13	A staircase polygon	23
2.14	Polygons requiring $\lfloor n/4 \rfloor$ edge guards (Shermer [1992])	32
2.15	The two polygons requiring $\lfloor (n+1)/4 \rfloor$ edge guards (Shermer [1992])	32
2.16	A polygon and its visibility graph	35
2.17	A multiply connected simple polygon (a simple polygon with holes)	38
2.18	Point-point visibility	39
2.19	Placing a vertex guard	40
2.20	Placing the next vertex guard	41
2.21	A vertex guard set for the example polygon	42

2.22	Part of a <i>cover</i> with convex polygons (note the overlapping of the convex polygons)	43
2.23	A minimum partition with convex polygons	44
2.24	An example of placing a minimum number of maximal axial lines	45
2.25	An example of stabbing boxes in two-dimensions – no stabbing line exists in this case	46
2.26	A subset of the minimum partition where it is necessary to determine if an axial line can be placed to cross the adjacencies between the convex polygons	47
2.27	The relationship between attempting to place an axial line through a number of adjacencies and edge to edge visibility in a polygon	48
2.28	Relating L_k visibility to axial lines	49
2.29	A traditional art gallery	50
2.30	“Ray guarding” a traditional art gallery with orthogonal ray guards	51
2.31	A placement of axial lines of arbitrary orientation in a traditional art gallery	52
2.32	The edge to edge visibility algorithm [Avis <i>et al.</i> , 1986] – totally facing edges	57
2.33	The edge to edge visibility algorithm – chain $C(y, u)$ cutting through quadrilateral $Q(u, v, x, y)$, no visibility is possible	58
2.34	An example of the edge to edge visibility algorithm – the input polygon and $Q(u, v, x, y)$	58
2.35	An example of the edge to edge visibility algorithm – the reduced chains	59
2.36	An example of the edge to edge visibility algorithm – the inner convex hulls	59
2.37	A complete grid of size 4	61
3.1	A simple configuration showing the two different problems	66
4.1	A simple configuration showing the two different problems	70
4.2	A configuration where the solution is not unique	71
4.3	Creating a biconnected planar graph	74
4.4	An example of a “triangle graph”, T_j	75
4.5	An example of adding triangle graphs to a graph to make it biconnected [(a) The original graph, v_1 and v_2 are cut vertices. (b) Graph with T_1 added, v_1 is still a cut vertex. (c) Final biconnected graph]	76
4.6	Creating a “stick” diagram	78
4.7	An example of the transformation of a biconnected planar graph to a stick diagram	80
4.8	The Canonical Unit which produces two choice axial lines (shown as dashed lines)	82

4.9	Joining the upper choice axial lines of two choice units	83
4.10	Joining the upper choice axial line of one unit to the lower choice axial line of the next unit	83
4.11	An example of converting a stick diagram to a collection of adjacent rectangles	85
4.12	The algorithm for determining the adjacencies between the rectangles	87
4.13	A configuration of adjacent orthogonal rectangles	89
4.14	Functions used in the algorithms in this chapter	91
4.15	Determining all possible orthogonal axial lines – Phase 1 part a . . .	91
4.16	Determining all possible orthogonal axial lines – Phase 1 part b . . .	92
4.17	Finding the essential lines – Phase 2	94
4.18	A configuration of adjacent orthogonal rectangles	94
4.19	Removing Redundant lines – Phase 3	95
4.20	Resolving the issue of choice – Phase 4	96
4.21	An example where the heuristic algorithm would not return an op- timal solution	98
4.22	A configuration in which there are $O(n^2)$ adjacency crossings	99
4.23	A configuration which forces the algorithm to extend $O(n)$ lines backwards	100
4.24	A “chequerboard” collection of rectangles	103
4.25	A “chequerboard” with holes	104
4.26	A simple configuration of rectangles with a rectangular union	105
4.27	Projecting Adjacencies onto Intervals on the line L	105
4.28	A simple configuration of rectangles that can be used in the produc- tion of an interval graph	106
4.29	A simple configuration of rectangles that cannot be used in the pro- duction of an interval graph	106
4.30	An example of a chain	107
4.31	An example of a tree of rectangles	108
4.32	A case where more axial lines than necessary are generated	109
4.33	The algorithm for placing orthogonal axial lines in chains of orthog- onal rectangles – Stages 0 to 2	110
4.34	The algorithm for placing orthogonal axial lines in chains of orthog- onal rectangles – Stage 3	111
4.35	The algorithm for placing orthogonal axial lines in chains of orthog- onal rectangles – Stage 4	112
4.36	The algorithm for placing orthogonal axial lines in chains of orthog- onal rectangles – Stage 5	113
4.37	A chain of orthogonal rectangles showing the forward and reverse lines and the final maximal lines.	115

4.38	The algorithm for placing orthogonal axial lines in trees of orthogonal rectangles – Stages 0 to 2	117
4.39	The algorithm for placing orthogonal axial lines in trees of orthogonal rectangles – Stage 3	118
4.40	The algorithm for placing orthogonal axial lines in trees of orthogonal rectangles – Stage 4	119
4.41	The algorithm for placing orthogonal axial lines in trees of orthogonal rectangles – Stage 5	120
4.42	An example of placing orthogonal axial lines in a tree of orthogonal rectangles	123
4.43	A tree with $n/2$ leaves and height also $n/2$	125
4.44	Choice introduced where each rectangle has at most 2 left and 2 right neighbours	127
4.45	An example of choice in configuration where each rectangle has at most 2 left and 2 right neighbours	128
5.1	An example of the problem	131
5.2	The Canonical Choice Unit which produces choice axial lines with arbitrary orientation	133
5.3	Connecting the upper portion of one ccu to the lower portion of the next	134
5.4	Connecting the upper portions of two ccus	134
5.5	Possible rays from a “horn”	135
5.6	An example of converting a stick diagram to a collection of adjacent rectangles	136
5.7	Placing a arbitrary axial line in a chain of rectangles	140
5.8	Converting a chain of rectangles into an adjacency polygon	141
5.9	Converting an adjacency polygon into a reduced adjacency polygon	143
5.10	An example of using the algorithm extend lines into all neighbours	144
5.11	An example showing some lines which would not be generated	145
5.12	An example showing the lines which would be generated in a top-bottom and left-right manner	146
5.13	Longest chains – identifying “extreme” rectangles	147
5.14	The longest chain heuristic	148
5.15	The longest chain heuristic: A problem with the heuristic – there are redundant lines in the final set of lines	149
5.16	The longest chain heuristic: A second problem with the heuristic – all the adjacencies are uncrossed after all the extreme rectangles have been considered	150
5.17	Crossing one adjacency at a time: Cases which cause “kinking” in a chain of rectangles	151

5.18	Crossing one adjacency at a time – example input	153
5.19	Crossing one adjacency at a time – the first two passes of the algorithm	153
5.20	Crossing one adjacency at a time – a possible solution	154
5.21	Crossing one adjacency at a time: A problem with the heuristic – lines that don't extend far enough to the left	155
5.22	Crossing one adjacency at a time: A second problem with the heuristic – lines which only cross a single adjacency	155
5.23	Extending forwards and then backwards – the different stages of determining chains, modifying the chains and placing axial lines. . .	157
5.24	Extending forwards and then backwards – redundant lines can be generated.	158
5.25	A more general chain of rectangles	160
6.1	An example of placing axial lines to cross all of the adjacencies in a configuration of adjacent convex polygons	164
6.2	A chain of convex polygons	166
7.1	(a) A complete grid of size 4, (b) The corresponding complete urban grid of size 4	169
7.2	An example of partitioning the outer thoroughfares of a complete urban grid of size 4	171
7.3	Another partitioning of the outer thoroughfares of a complete urban grid of size 4 – other similar partitionings exist	171
7.4	Possible adjacencies which could occur in a corridor intersection. Case 1 – Through the intersection (3 convex polygons involved) Case 2 – Rectangular ending at the intersection (4 convex polygons involved) Case 3 – Diagonal ending at the intersection (4 convex polygons involved) Case 4 – L-shaped diagonal ending at the inter- section (4 convex polygons involved)	173
7.5	A complete partitioning of a complete urban grid of size 4	174
7.6	The axial map for a complete urban grid of size 4	175
7.7	The axial map for a complete urban grid of size 4	176
7.8	An example of minimally partitioning a simple urban grid	177
7.9	178
7.10	An example of minimally partitioning a simple urban grid using only diagonal adjacencies at corners	178
7.11	An example of limited choice in placing axial lines in a simple ur- ban grid – only 3 of the dashed lines are necessary	179
7.12	An example of the limited choice in placing axial lines in a simple urban grid resulting in a cycle of choice axial lines – only 2 of the dashed lines are necessary	180

7.13 An example of placing an axial line to cross a single diagonal adjacency in two thoroughfares in a simple urban grid 181

7.14 An example of choice in placing axial lines in a general urban grid – only 3 of the dashed lines are necessary 182

7.15 An example of a deformed urban grid 184

7.16 A partition of a deformed urban grid 185

7.17 Partitioning a deformed urban grid – The description of the input into the algorithm and the functions used in the algorithm 186

7.18 Partitioning a deformed urban grid 187

7.19 A portion of a deformed urban grid – to illustrate the algorithm . . . 189

7.20 A partial solution from the algorithm to partition a deformed urban grid 190

7.21 Placing the axial lines to cross the adjacencies in a partitioned deformed urban grid 192

List of Tables

4.1	Experimental results	102
-----	--------------------------------	-----